OSIA USER MANUAL

The SWS IA Developer Team

Compiled by Rik Huygen (Rik.Huygen@ster.kuleuven.ac.be)
Edited by Aprajita Verma (verma@mpe.mpg.de)

Table of Contents

Contents

I OSIA basics 11

1 WHATIS OSIA 11

1.1 Introduction . . . 11
1.2 OSIA and IA^3 . . . 11
1.3 OSIA and ISAP . . . 11
1.4 What is new in OSIA version 4.0 . . . 11
1.5 What is new in OSIA version 3.0 . . . 12
1.6 What is new in OSIA version 2.0 . . . 12
1.7 OSIA Documentation . . . 12
1.8 Publication Guidelines . . . 13
1.9 The OSIA functionalities . . . 13

2 HowTo install OSIA on your system 17

2.1 Requirements . . . 17
2.2 Downloading . . . 17
2.3 Unpack it . . . 17
2.4 Setup . . . 18
2.5 More information and reporting problems . . . 18

3 HowTo Report a Software Problem 19

3.1 Introduction . . . 19
3.2 How to report . . . 19

4 HowTo read/write ISO/SWS data and calibration files 20

4.1 Introduction . . . 20
4.2 Reading the standard data products . . . 20
4.2.1 Reading ISO-SWS ERD with read_ferd . . . 20
4.2.2 Reading ISO-SWS SPD and AAR . . . 21
4.3 Reading calibration files . . . 21

4.4 Writing standard data products and calibration files . . . 22

5 HowTo use the right calibration files 23

5.1 Introduction . . . 23
5.2 The default set of calibration files . . . 23
5.3 Using the `cal_select` tool . . . 23
5.4 Summary of commands related to calibration files . . . 24
5.4.1 Commands related to the time dependency . . . 25
5.4.2 Commands related to the `cal_select` tool . . . 25
5.4.3 Other useful commands related to calibration files . . . 25

6 HowTo understand and manipulate ISO/SWS data structures 27

6.1 Introduction . . . 27
6.2 Structure description . . . 28
6.2.1 Type . . . 28
6.2.2 History tag . . . 28
6.2.3 Header . . . 28
6.2.4 Data . . . 29
6.3 Creating and manipulating IA^3 data structures . . . 29

7 HowTo understand and manipulate flags / statuswords 30

7.1 Status words . . . 30
7.2 Flag words . . . 32
7.3 Decoding . . . 32
7.3.1 Status words . . . 33
7.3.2 Flag words . . . 33
7.4 Manipulate . . . 34

8 HowTo link ISAP to you OSIA session 35

8.1 Introduction . . . 35
8.2 Procedure . . . 35
8.3 Fixes . . . 35

II OSIA Data Processing **36**

9 HowTo get a quicklook at your observation **36**

9.1 Getting help with routines . . . 36
9.2 Standard Plots and Interactive tools . . . 36
9.3 Scan Tables . . . 37
9.4 Quicklook . . . 38
9.5 Plot_phot . . . 39
9.6 Quality Check . . . 39

10 HowTo process SWS data from ERD to SPD **41**

10.1 Accessing data . . . 41
10.1.1 Select calibration files . . . 41
10.2 Derive Standard Processed Data - DSPD . . . 41
10.2.1 Valid data range - DRANGE (CAL04) . . . 42
10.2.2 Electronic zero level - MIDBIT (CAL02A) . . . 42
10.2.3 High-pass filter - ACCOR (CAL02,03) . . . 42
10.2.4 Reset pulse after effect - PULSE_SHAPE (CAL03,05) . . . 43
10.2.5 Detector cross-talk - CROSSCOR (CAL01) . . . 43
10.2.6 Slope calculation - SLOPE (CAL03,05,06) . . . 44
10.2.7 Assign wavelengths - WAVE (CAL12,16(a,b,c,et),18,23,24) . . . 44
10.2.8 Write the Standard Product Data . . . 44
10.3 Calfiles used in DSPD . . . 45

11 HowTo process SWS data from SPD to AAR **46**

11.1 Introduction . . . 46
11.1.1 Standard pipeline . . . 46
11.2 OSIA DAAR processing steps . . . 46
11.2.1 Order of commands and basic syntax . . . 47
11.2.2 Select calibration files . . . 47
11.2.3 Creating the SPD IDL structure . . . 47
11.2.4 SPD scantable . . . 47
11.2.5 SPD plotting . . . 48

11.3 OSIA DAAR routines 48
11.3.1 Reproducing pipeline reduction in OSIA 48
11.3.2 Transient Correction 49
11.3.3 Dark Current Subtraction 49
11.3.3.1 dark 49
11.3.3.2 dynadark (dark) 49
11.3.3.3 dark_inter (dark) 49
11.3.4 updown 50
11.3.5 Responsivity Calibration 50
11.3.5.1 respcal 50
11.3.5.2 resp_inter (respcal) 50
11.3.5.3 fringes (respcal) 50
11.3.6 Flux Calibration 50
11.3.6.1 fluxcon 50
11.3.6.2 plot_phot (fluxcon) 51
11.3.7 Velocity Correction 51
11.3.8 Extracting AAR data 51
11.3.9 Additional interactive processing routines 52
11.3.9.1 mask_inter 52
11.3.9.2 patch_spd 53
11.4 Offband 4 data 53
11.5 AAR processing 53
11.6 SPD manipulation routines 53
11.7 Pipeline warnings and error messages 54
11.7.1 DAAR shell 55
11.7.2 DARK (including DYNADARK) 55
11.7.3 FLUXCON 55
11.7.4 RESPCAL 56
11.7.5 VELCOR 56
11.7.6 EXTRACT_AAR 56

12 HowTo perform dark subtraction with DARK_INTER **57**

12.1 Introduction . . . 57
12.2 How to use DARK_INTER . . . 58
12.2.1 The DARK_INTER main window . . . 58
12.2.2 The 'plot' window . . . 60
12.2.3 The 'revise darks' window . . . 61
12.2.3.1 Revise all darks of current band . . . 62
12.2.3.2 Revise individual dark measurements . . . 62
12.2.4 The 'method' window . . . 64
12.2.5 Perform dark subtraction . . . 64
12.3 Saving results . . . 67

13 HowTo make interactive updown correction 68
13.1 Introduction . . . 68
13.2 HowTo make updown correction for SWS01 observations . . . 68
13.3 HowTo make updown correction for SWS02 observations . . . 72
13.4 HowTo make updown correction for SWS06/07-SW observations . . . 75

14 HowTo do RSRF correction interactively to minimise fringe residuals 76
14.1 Introduction . . . 76
14.2 RSRF Features . . . 76
14.2.1 Observed Fringes . . . 76
14.2.2 RSRF Fringes . . . 76
14.3 RSRF correction in OSIA . . . 77
14.4 RESP_INTER . . . 77
14.4.1 General use . . . 77
14.4.2 Implementation . . . 77
14.4.3 Particular cases . . . 78
14.4.3.1 Fast AOT-1 observations . . . 78
14.4.3.2 Low signal data . . . 78
14.4.3.3 Confused spectral regions . . . 78
14.4.3.4 AOT-2 data . . . 78
14.5 Future Developments . . . 79

15 HowTo minimise fringe residuals in SWS data **83**

15.1 Introduction . . . 83

15.2 OSIA modules . . . 83

15.3 RESP_INTER . . . 84

15.3.1 Default operation . . . 85

15.3.2 Reduction examples . . . 85

15.3.2.1 Standard calls . . . 86

15.3.2.2 Spectral confusion . . . 86

15.3.2.3 Using parameter input . . . 87

15.4 FRINGES . . . 87

15.4.1 Advantages over FFT . . . 88

15.4.2 Using FRINGES . . . 88

15.4.2.1 Up and down scans . . . 90

15.4.2.2 Limiting the cycle range . . . 90

15.5 AARFRINGE . . . 92

15.5.1 AOT-2 scans . . . 93

16 HowTo reduce a grating AAR (within the SWS OSIA Environment) **94**

16.1 Introduction . . . 94

16.2 Available Routines . . . 94

16.2.1 aarfilt . . . 94

16.2.2 cleanstruct . . . 95

16.2.3 combine . . . 95

16.2.4 select . . . 95

16.2.5 sigclip . . . 96

16.2.6 sws_rebin . . . 96

16.2.7 sws_flatfield . . . 96

16.2.8 sws_grid . . . 96

16.3 Pitfalls . . . 96

16.3.1 SIGMA-CLIPPING . . . 96

16.3.1.1 AOT-1 . . . 97

16.3.2 FLAT-FIELDING . . . 97

16.3.2.1 Flat-field order . . . 97
16.3.2.2 Reference spectrum . . . 97
16.3.2.3 Offset or gain . . . 97
16.4 AOT Specifics . . . 98
16.4.1 AOT-1 . . . 98
16.4.2 AOT-2 . . . 98
16.4.3 AOT-6 . . . 98
16.4.4 AOT-7 SW grating data . . . 98
16.5 Future Developments . . . 98

17 HowTo reduce an SWS AAR AOT7 **99**
17.1 Introduction . . . 99
17.2 Flatfielding . . . 99
17.3 Clipping . . . 99
17.4 Rebinning . . . 100
17.5 Troubleshooting . . . 100

18 HowTo reduce data using the LOW_SIGNAL analysis tool **101**
18.1 Reduction tool . . . 101
18.1.1 Type of data . . . 101
18.1.2 Data input/output . . . 103
18.1.2.1 Information . . . 103
18.2 LOW_SIGNAL selection options . . . 103
18.2.1 DATA selection . . . 103
18.2.2 DET, detector selection . . . 103
18.2.3 STDEV, standard deviation clipping . . . 104
18.2.4 SIGNAL, signal level clipping . . . 104
18.2.5 FLATFIELD, flat-fielding (SWS_FLATFIELD) . . . 104
18.2.6 FRINGES, remove fringe residuals (FRINGES) . . . 104
18.2.7 DRIFT, drift filter (DRIFT_FILTER) . . . 105
18.2.8 CORREL, correlated noise filter (CORR_NOISE) . . . 105
18.2.8.1 Application of correlated noise filter . . . 105

18.2.9 SIGCLIP, sigma clipping (SIGCLIP) . . . 105
18.2.10 NEXT filter . . . 106
18.2.11 SMOOTHING . . . 106
18.2.12 Line fits (GAUSS_FIT or LORENTZ_FIT) . . . 106
18.2.13 Lineflux (LINEFLUX) . . . 106
18.3 Description of a spectral scan . . . 106
18.3.1 AOT-2 scans . . . 106
18.3.2 Resolution . . . 106
18.3.2.1 Spectral Resolution . . . 108
18.3.2.2 LVDT Resolution . . . 108
18.3.2.3 Detector Resolution . . . 109

19 HowTo obtain high resolution data for AOT1 observations 110
19.1 Introduction . . . 110
19.2 Implementation . . . 110
19.3 Examples . . . 110

III Pointing 111

20 HowTo determine the size/orientation of the SWS slit on the sky 111
20.1 Introduction . . . 111
20.2 Aperture definition . . . 111
20.3 Slit Sizes & Offsets . . . 113
20.4 How slit size/pointing can affect observations . . . 117

21 HowTo correct mispointed observations and Solar System Objects observations 118
21.1 Introduction . . . 118
21.2 The beam profile calibration files . . . 118
21.3 The routine POINT_CORR . . . 118
21.4 Limitations . . . 119

IV Miscellaneous 121

22 HowTo access SWS errors **121**

22.1 Introduction . . . 121

22.2 Errors in SPD and AAR . . . 121

22.2.1 Error encoding in the aar . . . 122

22.3 Using errors in OSIA . . . 122

22.3.1 use of the aar error in OSIA . . . 123

22.3.1.1 sws_rebin . . . 123

23 HowTo measure a line flux in IA **124**

23.1 Introduction . . . 124

23.2 Direct integration using GETLINEFLUX . . . 124

23.3 Nonlinear least-square fit using MULTI_FIT . . . 125

V Reference list **126**

VI Document change log **127**

Part I

OSIA basics

1 WHATIS OSIA

1.1 Introduction

The ISO data processing software can be split into five major parts: the pipeline, composed of the instrument dependent pipeline (designed and developed by the instrument teams) and the instrument independent pipeline (designed and developed by ESA); IA^3 (used internally by SWS); the ISO Spectral Analysis Package (ISAP); and the Observers SWS Interactive Analysis (OSIA).

OSIA is a version-controlled subset of IA^3 which contains modules which are usable for advanced data processing for scientific purposes.

1.2 OSIA and IA^3

IA^3 installations have to consider different platforms (directory structures, setup procedures etc.) and all the hardware specifics.

The embedding of the configuration control system required an installation and registration at the CoCo server and sites are required to name a manager who takes the responsibility to maintain this IA^3 system. Although individual IA^3 modules are version controlled, the IA^3 software system as a whole is not.

Contrary to that, OSIA is strictly version controlled which allows referencing on certain software versions. One of the major design aspects was to keep OSIA simple, platform independent and easy to install. Thus OSIA contains only pure IDL modules and FITS calibration files.

1.3 OSIA and ISAP

To make a clear separation between OSIA and ISAP, OSIA will work on the Edited Raw Data (ERD, since version 2.0), Standard Processed Data (SPD) and Auto-Analysis Product (AAR) levels whereas ISAP consists of processing tasks based solely upon the AAR level. OSIA was developed for data reduction (ERD to AAR level) and includes functionalities for post-AAR analysis. ISAP was created for AAR product analysis rather than reduction and is complementary to OSIA. Thus OSIA and ISAP are ideal partners for scientific data processing. Special effort has been taken to make both software packages work in a single environment (one session together).

1.4 What is new in OSIA version 4.0

1 Version 4.0 is based on PIPELINE version 10.1

2 High resolution AOT-1 processing routines are now available.

3 RSRF improvements to bands 3A, 3C and 3D. CAL25_3a_060 are now the default calibration files. CAL25_3c_060 and cal25_3d_060 are available but are *not defaults*. `CAL_SELECT` should be used to apply the latter two new RSRFs.

4 Improvement in the pulse-shape correction involved in the DSPD step, giving a slight improvement in some cases.

5 Several small changes made and bugs fixed.

6 Inclusion of perl script osia_man with the distribution.

1.5 What is new in OSIA version 3.0

1 Version 3.0 is based on PIPELINE version 10.1

2 Improved error propagation.

3 Pointing and SSO tracking correction.

4 Coherent masking and flagging between PIPELINE modules and OSIA.

5 Calfile selection tool.

6 Improved tools for defringing.

7 Improved RSRF and Flux Calibrations for bands 2A and 2C.

1.6 What is new in OSIA version 2.0

1 Version 2.0 is based on PIPELINE version 9.5.

2 Version 2.0 supports data processing from ERDproducts onwards. As one consequence the complete set of instrument specific PIPELINE is available. It is possible to execute the PIPELINE step by step, visualize intermediate results or use interactive modules instead

3 Off-band data (band 4A, 4C and 4D) are supported

4 The error computation has been improved - see "The error propagation of the SWS PIPELINE" available at http://sws.ster.kuleuven.ac.be/documents

5 Wavelength computation has been improved. The wavelength shifts are analyzed and implemented by polynomes. Post Helium Phase and Instrument Level Test wavelength calculation has been implemented

6 Calibration files : Beam Profile for pointing and jitter correction

7 Calibration files : Beam Profile for extended source flux correction

8 Numerous small improvements

1.7 OSIA Documentation

Prior to beginning OSIA processing, it is recommended that the user familiarises themselves with the SWS instrument through the ISO and SWS instrument handbooks available as part of the ISO Explanatory Library (*see the following URL* http://www.iso.vilspa.esa.es/users/expl_lib/SWS_top.html).

OSIA documentation is broadly split into three main components: HowTos, dc-documents and supporting documentation available via the internet.

1 **HowTos**

The OSIA manual and the web-page help consists of a series of HowTos. HowTos are easy to follow documents containing a description of individual steps (sometimes interactive) used within the reduction of SWS data. They encompass and explain the stages within the data reduction so that the user may be able to track the reduction of the data at all stages. Certain automated pipeline modules can be replaced by interactive steps, these are described in more detail in separate HowTos. In the description of the processing steps, pipeline modules which may be replaced by an interactive version, you will be directed to the relevant HowTo detailing this interactive module. Throughout the text the reader will be referred to other HowTos within the OSIA manual (and at the OSIA web pages).

2 **dc-documents**

In addition to HowTos, help is available online using

```
OSIA> widget_olh
```

or the versatile shell command `osia_man`. A summary of the usage and syntax is printed using

```
YOURHOST> osia_man
```

These two commands describe the basic OSIA functionalities (listed in the following section) and calibration files. Throughout the text the online help documents are referred to as 'dc-documents' or 'online help'.

3 **Supporting documentation**

A wide range of supporting documentation is available as part of the ISO Explanatory Library (*see the following URL* http://www.iso.vilspa.esa.es/users/expl_lib/SWS_top.html) or on the SWS instrument team document resources (*see the URLs* http://sws.ster.kuleuven.ac.be/documents, http://www.sron.rug.nl/didac or http://www.mpe.mpg.de/ISO/isosdc.html).

Where internet-available documents are cited, full web-locations are given.

1.8 Publication Guidelines

If you use OSIA to reduce your SWS data please write in the acknowledgements:

"OSIA is a joint development of the SWS consortium. Contributing institutes are SRON, MPE, KUL and the ESA Astrophysics Division."

If you use IA^3 to reduce your SWS data please write in the acknowledgements:

"IA^3 is a joint development of the SWS consortium. Contributing institutes are SRON, MPE, KUL and the ESA Astrophysics Division."

1.9 The OSIA functionalities

The OSIA functionalities can be grouped as follows:

1 Installation modules
Modules which are necessary to install the system :

`OSIA_SETUP.BASH`	setup osia environment on UNIX computers (bash)
`OSIA_SETUP.CSH`	setup osia environment on UNIX computers (csh)
`OSIA_SETUP.VMS`	setup osia environment on VMS computers
`OSIA_STARTUP.DAT`	OSIA IDL startup file

2 DSPD pipeline modules (from version 2.0 on)
These modules run the pipeline steps between SPD and AAR separately or together :

DSPD	Run complete pipeline DSPD step
DRANGE	Flag invalid data
MIDBIT	Correct for zero voltage point
ACCOR	Correction of the integration ramp for RC time constant
PULSE_SHAPE	Correction of Reset pulse effects
CROSSCOR	Detector cross-talk correction
SLOPE	Slope calculation inclusive glitch handling
WAVE	Wavelength calculation

3 DAAR pipeline modules
These modules run the pipeline steps between SPD and AAR separately or together :

DAAR	Run complete pipeline DAAR step
ANTIMEM	skip memory affected samples for dark subtraction
DARK	calculate and subtract dark currents like in the pipeline
RESPCAL	responsivity calibration
FLUXCON	absolute flux calibration
VELCOR	Correct wavelength for velocity
EXTRACT_AAR	Put valid data into a flat AA structure

4 Tools to get an overview

MAKE_SCTBL	make a scan table
PRINT_SCTBL	print/display a scan table
QUICKLOOK	creates an IDL script file, which allows easily to preview/print lines/ranges of AOT S02 or S06
LIST_TIMEDEP	List calibration file time dependency file

5 Tools to treat status and flags
Decoding etc. of status and flag words :

DECODE_FLAG	Decode flag word
DECODE_STATUS	Decode status word
MAKE_FLAG	Make flag entry
MAKE_STATUS	Make status entry
TEST_FLAG	Test flag word
TEST_STATUS	Test status word

6 Header modules
Modules to extract data product header information :

LIST_HEADER	List structure-product header
XMOD_HEADER	Modify structure-product header

7 SWS data product I/O
Tools to access most SWS data products :

CAL_SELECT	Select calibration files as default
READ_CAL	Read calibration files (ignore TEST / BASE)
READ_FAAR	Read SWS AAR product
READ_FAOCS	Read SWS AOCS file
READ_FEOHI	Read SWS EOHI product
READ_FERD	Read SWS ERD file
READ_FGEHK	Read SWS GEHK file
READ_FGLL	Read SWS Glitchlist file
READ_FIIPH	Read SWS IIPH file
READ_FIRPH	Read SWS IRPH file
READ_FSPD	Read SWS SPD product
READ_FSSTA	Read SWS SSTA product
READ_TIMEDEP	Read SWS calibration file time dependency file

8 Common FITS I/O

WRITE_FITSTABLE	Write a structure to a FITS table
READ_FITS_KEY	Read FITS header keyword from structure
WRITE_FITS_KEY	Write FITS header keyword to structure
COMPARE_FITSKEY	Compare FITS entries of structures
DELETE_FITSKEY	Delete FITS header keyword from structure

9 Interactive pipeline steps

Modules which fulfil the task of the pipeline, but can be used in a highly interactive manner to achieve a better result :

GLITCH_VIEWER	Interactive Glitch recognizion, removal and glitch tail correction
DARK_INTER	Interactive dark subtraction
UPDOWN_INTER	Interactive up/down scan correction (AOT 2,6,7)
RESP_INTER	Interactive Responsivity Calibration via shifting and smoothing or enhancing of the RSRF

UPDOWN1_INTER, the updown scan correction for AOT1 observations, is not yet implemented. This tool requires data from pre-OSIA data processing which are not yet available.

10 Masking tools

Visualize the SPDdata product and remove invalid data points :

MASK_INTER	Interactive visualizing and masking of SPD data points

11 Selecting & Combining

SELECT	select records of an IA structure into new (shorter) one
COMBINE	combine two spectra

12 Patching SPDs

PATCH_SPD	remove jumps, slopes, glitchtails on SPD level

13 Defringing

Fringes	remove fringes (especially from band 3) on SPD or AAR level (from version 2.0 on)

14 Flatfielding

A very extensive tool to flatfield SWS spectrum :

SWS_FLATFIELD	Flatfield all separate det scans per AOT band, line and scan

15 Clipping

Standard sigma clipping tools with a lot of options :

SIGCLIP	Perform sigma clipping on an AAR or SPD
STDEVCLIP	clip potential glitches indicated by high stdev or outlier flux

16 Rebinning

A very extensive tool to rebin SWS spectrum :

SWS_REBIN	Rebin an AAR onto a predefined grid or on a grid defined by a const. resolution or const. binsize
AARFILT	extract a filtered spectrum for a line out of an AAR

17 Pointing correction

POINT_CORR	Pointing and SSOP tracking correction
APER_PLOT	Plot an SWS aperture projected on the sky for a given observation

18 Plotting

Standard plotting of all relevant data products, SWS aperture on the sky and internal checks :

PLOTERD	plot ERDproduct
PLOTSPD	plot SPDproduct
PLOTAAR	plot AARproduct
PLOT_FLAG	Plot the flags per detector below a SPD plot
PLOT_FULLFLAG	Plot the status below the entire ERD/SPD/AARplot
PLOT_FULLSTATUS	Plot the status below the entire ERD/SPD/AARplot
PLOT_PHOT	Plot photometric checks
PLOT_STATUS	Plot the status below the entire ERD/SPD/AARplot

19 Calibration G files

STIMEDEP_BASE	Calibration time dependency file
CAL01_xx	electronic Cross talk matrix
CAL02A_xx	Calibration file for the 0 Volt point in bitrange
CAL02B_xx	Calibration file for the effects of the reset pulse aftermath
CAL02_xx	Calibration file for RC timeconstants
CAL03_xx	Calibration file reset cutout length
CAL04_xx	Calibration file for amplifier limits
CAL05_xx	Calibration file for switchable gain values
CAL06_xx	Glitch reject levels
CAL07_xx	Noise values
CAL08A_xx	Grating Calibration Spectrum
CAL08_xx	Grating Calibration Spectrum
CAL09_xx	LVDT-angle relation for QLA
CAL11_xx	Calibration file for fixed FP transmission peaks for FM
CAL12_xx	Calibration file for position - gap relation of scanning FPs
CAL13_xx	Flux calibration at key wavelength
CAL16A_xx	Aperture Offsets
CAL16B_xx	Element Offsets
CAL16C_xx	Grating Constants
CAL16D_xx	Telescope area
CAL16E_xx	Polynomial coefficients
CAL18_xx	Calibration file for wavelength dependent FP gaps of FM
CAL19_xx	Full Width Half Max per AOT
CAL21_1_xx	Dark current for reset time 1
CAL21_2_xx	Dark current for reset time 2
CAL21_4_xx	Dark current for reset time 4
CAL21_8_xx	Dark current for reset time 8
CAL22_xx	Response to internal sources
CAL23_xx	Lists for each of the 6 detector bands the wavelength limits
CAL24_xx	Wavelength limits of apertures
CAL25_band_xx	Calibration file for relative spectral response
CAL35_xx	Beam Profile for pointing and jitter correction (from version 2.0 on)
CAL36_xx	Beam Profile for extended source flux correction (from version 2.0 on)
CAL40_xx	Scanner stops

19 Common tools

Various tools to support data reduction :

DECODE_AA_ERROR	Decode the error within the AA product (from version 2.0 on)
GET_AOT_NUMBER	Get AOT number from ERD/SPD or AAR header
GET_AOT_SPEED	Get AOT speed for SPD
GET_APERTURE	Get aperture number from status word
GET_ORDER	Determine grating order from flag word
GETLVDT	Calculate LVDT and grating angle from wavelength
RESOLUTION	get grating resolution
SHOWSTRUCT	show contents of a structure
MULTI_FIT	a universal fitting tool (from version 2.0 on)

20 Submodules

There are quite a number of submodules which are called by the main functionalities.

2 HowTo install OSIA on your system

OSIA (Observers SWS Interactive Analysis) is a subset of the SWS Interactive Analysis package (IA^3) and provides all necessary modules for advanced SWS data processing. This HowTo describes where this free software package can be obtained from and how to install it.

2.1 Requirements

OSIA runs on VMS and UNIX systems. OSIA is coded in IDL [1] and requires IDL version 5.0x for VMS systems and version 5.0 or higher for UNIX systems for execution.

IDL is a complete computing environment for the interactive analysis and visualization of data (*see the following URL* http://www.rsinc.com).

2.2 Downloading

You can download the latest version of OSIA from :
http://www.iso.vilspa.esa.es/archive/software/ or
http://sws.ster.kuleuven.ac.be/osia/

The software is distributed as a gzipped tar file `osia_vv.tar-gz` (where vv corresponds to version number).

2.3 Unpack it

Change to the directory where you want to put the directory containing the OSIA installation.

Unzip the file	:	gzip -d osia_vv.tar-gz
Untar the file	:	tar -xvf osia_vv.tar

This will create the directory osia_vv containing the entire OSIA software suite. From now on we will refer to this directory as OSIA_DIR.

OSIA_DIR will contain the IDL startup file for OSIA - `osia_startup.dat`.

Five sub directories will be created.

`src`	:	this directory contains OSIA IDL program code
`cal`	:	this directory contains the FITS calibration files
`doc`	:	this directory contains help files and documentation
`scripts`	:	this directory contains the perl script for the online manual `osia_man`
`data`	:	this directory contains ephemeris data for each ISO revolution for SSO tracking observations

For VMS systems it is necessary to reconfigure the file attributes of the FITS calibration files :

```
set file/attr=(lrl:2880,rfm:fix,rat:none) OSIA_DIR:[CAL]*.fits;*
```

[1] IDL is a registered trademark of Research systems inc.

2.4 Setup

To set the necessary parameters use your favorite editor to edit : `osia_setup.vms` on VMS; `osia_setup.bash` or `osia_setup.csh` on UNIX according to your preferred shell or operating system.

Edit the OSIA_DIR variable in these setup files and point it to your installation directory.

NOTE: The OSIA setup changes the IAS_ARC environment variable which might conflict with parallel IA3 sessions !

Make sure that the operating dependent startup file is called when you log into your system. This enables users to start OSIA just by typing 'osia' at the prompt and have fun...

2.5 More information and reporting problems

Every OSIA release comes with a postscript USER MANUAL (OSIA_USER_MANUAL_vv.PS , in the doc directory). For up to date documentation see one of the download WEB sites.

KULeuven administers the distribution of SWS software and documentation. Users may keep themselves informed about OSIA and issue problem reports via their WEB site
(*see the following URL* `http://sws.ster.kuleuven.ac.be/osia/`).

3 HowTo Report a Software Problem

This HowTo describes how to report software problems and programming suggestions to the SWS programming team.

3.1 Introduction

Bug reports and programming suggestions are handled by the SWS programming team in much the same way as a *Software Problem Report* or for short, SPR. For IA^3, we have a configuration controlled SPR system which is used internally by the consortium. For OSIA we provide a Web interface for bug reports on our OSIA Homepage (*see the following URL* http://sws.ster.kuleuven.ac.be/osia/).

Before reporting a problem or making any suggestions, please make sure you have checked all the other sources of information (OSIA User Manual, HowTo's, FAQ's) to see if the problem is described there.

3.2 How to report

The Web page for reporting problems with OSIA will present you with a form to be filled. This form is sent to the software maintainers for evaluation.

Please provide all requested information as accurately as possible. This will allow us to pin-point the problem and provide you with a solution in a reasonable amount of time.

The requested items are described in more detail below:

Name Tell us who you are and how we can reach you (email address).

Product name This will probably be OSIA, unless you have questions about a related product like, e.g., IDL.

Version number Since OSIA is version controlled, we can at any time track the errors you encounter in a specific version and check if it is reproducible. We therefore need the exact version number of OSIA.

IDL version Since some problems might be related to the version of IDL used, we like to know which version was used while the problem was encountered.

Environment The environment may be critical in determining and/or reproducing the reported problem. Please specify as much information as possible about your enviroment and related to the problem.

- Operating system and OS version, e.g., AXP VMS 6.2 or SPARC Solaris 2.5.1
- Hardware, memory and other resources, e.g., SPARC 10 with 256 Mbytes and a swap space of 512 Mbytes on external disk.

Problem Description Please note here how, when and where the problem occurred. Describe what you were doing, which commands you ran and on which data sets. Include the error message in this description.

Please submit the form once it is filled in. Your problem will then be handled as soon as possible. The OSIA SPRs will become available on the OSIA Problem Reports Web page.

4 HowTo read/write ISO/SWS data and calibration files

This HowTo describes how to read and write the ISO/SWS data products including the calibration files. The document does not describe the structure of ISO-SWS data products (*see the HowTo-document* HowTo `understand and manipulate SWS data structures`).

4.1 Introduction

The ISO-SWS standard data products are ERD, SPD, and AAR. These files represent steps in the standard pipeline processing. The ERD file contains edited raw data which is used as a starting point for the DSPD part of the pipeline. The result of this pipeline step is an SPD which in turn is used as the starting point for the DAAR part of the pipeline. The final result is the AAR which contains the —flux and wavelength— calibrated spectral data. For more information on the data structures themselves, refer to the HowTo document HowTo understand and manipulate SWS data structures. The structure of the FITS files is fully described in the ISO Data Product Document[15], the contents of the FITS files (which is copied into the data structures) is described in the ISO Handbook Volume V [11].

4.2 Reading the standard data products

OSIA provides routines to read the three standard data products. The following subsections will briefly describe how to use these routines in OSIA. For a more detailed description, please refer to the on-line documentation (dc-documents) for each routine accessible through the OSIA command `widget_olh` or the perl program osia_man (see the dc-documents `read_ferd`, `read_fspd` and `read_faar`).

> NOTE: File- and pathnames that are passed to OSIA functions should be specified in a format which is understood by the operating system.

4.2.1 Reading ISO-SWS ERD with read_ferd

When reading in an ERD there are actually four files to be read, i.e., the ERD itself and three auxiliary files that contain information about the status of the instrument and the satellite at the time of the observation. All these files are in FITS format.

To read an ERD use the `read_ferd` routine.

```
erd = read_ferd ('TTTTTTOO' [, path=<pathname>]
                            [, length=<number of seconds>]
                            [, skip=<even number of seconds to skip>]
                            [, /cd])
```

This routine expects the four files in the same directory —which is given by the `path=` argument— and assumes that all files have their standard names, i.e., `swerTTTTTTOO.fits`, `gehkTTTTTTOO.fits`, `sstaTTTTTTOO.fits` and `eohiTTT.fits`.

The first argument identifies the observation, where `TTTTTT` represents the TDT number of the observation and `OO` is the 'observation sequence number' (the number of the observation within your programme). The TDT number contains the revolution number in the first three digits and the target number (a counter which represents the n-th observation in the revolution) as the last three digits.

Unix Example :

```
erd = read_ferd('12300701',path='/home/mydir/dummydata')
```

VMS Example :

```
erd = read_ferd('12300701',path='[home.mydir.dummydata]')
```

If you are reading your data from the CD-ROM, you have to supply the `/cd` keyword and the path should point to the directory that contains the ERD. The filename extention on the CD-ROM is `.fit`.

4.2.2 Reading ISO-SWS SPD and AAR

The SPD and AAR products are contained in a single FITS file which makes the read routine much simpler:

```
spd = read_fspd (<filename>)
aar = read_faar (<filename>)
```

The `<filename>` should contain the full pathname of the SPD. If no pathname is given, `read_fspd` and `read_faar` tries to find the file in the current directory.

Unix Example :

```
spd = read_fspd('/home/mydir/dummydata/swsp12300701.fits')
```

VMS Example :

```
spd = read_fspd('[home.mydir.dummydata]swsp12300701.fits')
```

4.3 Reading calibration files

Calibration files are read with the OSIA function `readcal` as follows:

```
cal = readcal ('calnr' [, calfile=<custom cal filename>])
```

The `calnr` identifies the type of calibration structure. It can be the name of the calibration file, like `cal16b`, or only the number, like `25_1a`. `calnr` is case-insensitive.

If you want to read a calibration file other then the default file in the OSIA distribution, use the cal_select module (see HowTo use the right calibration files) or the `calfile` keyword and specify the full pathname to the custom calibration file.

For more information about calibration files and how to define which files will be used by OSIA please refer to HowTo use the right calibration files.

4.4 Writing standard data products and calibration files

The SWS interactive analysis software provides one interface for writing the standard data products (e.g. intermediate and final SPD, AAR or cal-files but not ERDs) as FITS files. The command syntax is as follows:

```
write_fitstable, <data structure>, <filename>
```

Where `<data structure>` is the name of a valid OSIA data structure. The output is written to `<filename>` in the current directory if `filename` doesn't contain a path. The extension `.fits` is *not* automatically appended and should be part of `filename`.

Please note that this is an IDL procedure, in contrast to the read routines which are functions, the arguments are not passed between brackets and there is no return value.

For more information about `write_fitstable`, refer to the on-line dc-documentation for `write_fitstable`.

5 HowTo use the right calibration files

The SWS data processing software OSIA uses a number of calibration files that contain a quantitative characterisation of different properties of the instrument. Some calibration parameters describe properties that changed during the course of the mission, because of drifts in the instrument or because of a change in the way the instrument was operated. The corresponding calibration files are therefore time dependent, which means there are different versions of the calibration files that are only valid for observations obtained during a certain period. This document explains how to use the tools that are provided for selecting the correct calibration files for your SWS data.

5.1 Introduction

The document will guide you through all the commands and steps to select and use the correct calibration files in your OSIA data processing.

There are ninety three calibration files for SWS. The format and contents of these files are explained in Appendix A of the ISO-SWS Handbook[11].

A number of these calibration files contain parameters that define instrument characteristics that are variable with time. These calibration files have several versions containing the parameter values that are valid for a certain period. The software automatically selects the correct version of the calibration file.

Documentation for each of the calibration files is accessible via the IDL command `widget_olh` under the button 'OSIA CAL'.

5.2 The default set of calibration files

By default, if you run the OSIA commands and tools, the system will automatically select all the calibration files used for the SWS data processing in OLP 10.1, i.e. the version of the Off-line-processing software at the ISO data centre used to process the data in the legacy archive.

The files are located in the calibration directory, which on UNIX is defined by the `$IAS_CAL` environment variable and on VMS by the logical `IAS_CAL`. This variable should be defined in your working environment if OSIA has been set up correctly. All calibration files are defined in the FITS format.

For time dependent calibration files the software selects the version for your SWS data based on (1) the UTC time in your raw data files and (2) the period defined in the `stimedep` file for which the time dependent calibration parameters are valid.

This period is defined as a table in the `stimedep` file located in the same directory as the calibration files.

5.3 Using the `cal_select` tool

If you want to use another version of a calibration file in your OSIA processing, you can do so by selecting that calibration file using the `cal_select` tool.

`cal_select` is a graphical tool that allows you to select a specific user defined calibration file. The GUI provides two choices for each calibration file (see Figure 1). You can either select `BASE` or `USER`. The `BASE` option means you want to use the default calibration file that the system automatically selects according to the timedep file. The `USER` option means you want to use a different calibration file that you manually select. This can be another version of the default calibration file that is available from OSIA, or a new file in which the parameter values have been changed. The format of the user supplied file should in any case be identical to the format of the default file.

If you start `cal_select` for the first time the `BASE` option is selected for all calibration files.

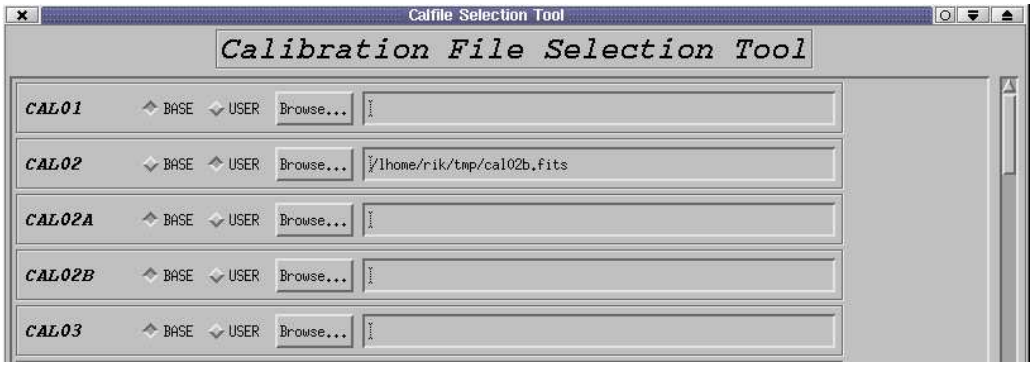
Calfile Selection Tool
Calibration File Selection Tool
CAL01 BASE USER Browse...
CAL02 BASE USER Browse... /lhome/rik/tmp/cal02b.fits
CAL02A BASE USER Browse...
CAL02B BASE USER Browse...
CAL03 BASE USER Browse...

Figure 1: The top part of the `cal_select` GUI tool.

To specify your own calibration file use the `Browse...` button. This will pop up a file selector box from which you can select the file. The `USER` option will then automatically be checked. Alternatively you can type in the *full pathname* of the file in the text box. In either case the `cal_select` tool does not check for the correctness of the filename.

Repeat this action for all calibration files that you want to be different from the default selection.

At the bottom of the GUI (see Figure 2) are a number of buttons for saving and loading the user selection, to clear and restore the selection and buttons to cancel or confirm the actions.

`Load` The `Load` button pops up a file selector box from which you can select a `cal_select` file that you have previously saved. The default filename for the saved settings is `cal_select.dat` and is located in your home directory.

`Save` The `Save` button pops up a file selector box to save the selected setting for the calibration files. The selection will be saved in a file that can be loaded at a later time from the GUI or with the command `load_cal_selection` (see section 5.4.2).

`Clear` The `Clear` button resets all selections to the default settings i.e. the `BASE` option.

`Restore` The `Restore` button can be used to cancel out all changes that you made since you started the `cal_select` tool. Even if you saved the selection in between the restore function will set you back to the situation where you started.

`Cancel` The `Cancel` button abandons all changes and exits the GUI without saving. A warning message will be issued.

`OK` The `OK` button exits the GUI and makes the new calibration file selection available to your current environment. Note that if you didn't save your selection it will be lost when you exit the OSIA session.

On exit the `cal_select` tool checks the changes that you have made and prints out a summary of these changes.

As long as you stay in your OSIA session the `cal_select` tool remembers the settings. If you want to keep the selection across OSIA sessions you need to save them from the `cal_select` GUI tool and restore (using the `LOAD` button) them in subsequent sessions as required.

5.4 Summary of commands related to calibration files

There are a number of commands that interact with the current calibration file selection that can be used from the OSIA command line. These commands will be described in this section.

Figure 2: The bottom part of the `cal_select` GUI tool.

5.4.1 Commands related to the time dependency

As has been said above there are a few calibration files for which the parameter values have been derived at different times during the ISO mission. The periods for which those parametsr are valid or optimal are kept in a table in the `stimedep` file. Under normal circumstances you do not need to modify this table, but you might want to take a look at the entries.

read_timedep This command is used to read the time depedency file for calibration files into a `TIMEDEP` structure in memory.

list_timedep Use this command if you want to inspect the contents of the specified `TIMEDEP` structure. This will list all calibration files sorted by their OLP CALG filename together with the calibration filename + version number, the UTC timekey in the format `YYDDDHHMMSS` and a comment. The timekey indicates from which point in time the calibration file is valid.

is_timedep Use this command to test if the specified variable is a `TIMEDEP` structure.

define_timedep This command is used internally to create an empty `TIMEDEP` structure in memory.

5.4.2 Commands related to the `cal_select` tool

There are also a number of commands for inspecting and handling the `CAL_SELECT` structure in memory.

list_cal_selection This command is equivalent to the command `cal_select, /list` and prints the current selection for each calibration file.

load_cal_selection This is equivalent to the `Load` button in the `cal_select` tool. This command is useful to restore a specific calibration file selection at the start of your session or during batch processing as it doesn't use the GUI and is not interactive.

save_cal_selection This is equivalent to the `Save` button in the `cal_select` tool.

5.4.3 Other useful commands related to calibration files

The following commands are listed here for completeness only. They might come out useful when developing your own procedures or functions within OSIA.

default_calfile Get the name of the default calibration file for a specified name.

define_cal Create an empty calibration structure in memory for the given calibration file.

get_cal_name Get the full pathname for a calibration file.

which_cal_name Return a valid cal filename for a start-of-observation time.

getcaldir Return the directory name for calibration files given a logical or environment variable.

is_cal Test if the specified variable is a calibration structure and return its name.

readcal Create and fill a calibration structure.

6 HowTo understand and manipulate ISO/SWS data structures

Within SWS interactive systems ISO SWS data products, calibration files and other data are represented as IDL data structures. It is possible to apply plain IDL commands on this data structures, but also a number of interactive tools to deal with data structures are provided by the SWS interactive systems.

6.1 Introduction

Within SWS interactive systems ISO SWS data products, calibration files and other data are represented as IDL data structures (*see the following URL* http://www.iso.vilspa.esa.es/manuals/HANDBOOK/sws_hb). Structures are collections of scalars, arrays, or other structures contained in a variable. Typical examples of a Standard Processed Data (SPD) structure and an Auto Analysis (AAR) structure with 100 entries are given below, which are the outputs from the tool `showstruct`.

```
OSIA > showstruct,spd
% Compiled module: SHOWSTRUCT.

(1)
   |----TYPE<Strng>(1)
   |-HISTORY<Strng>(1)
   |--HEADER<Strng>(1)
   |----DATA<Struc>(100)
                        |----DET<Struc>(52)
                                          |---WAVE<Float>(1)
                                          |---FLUX<Float>(1)
                                          |-OFFSET<Float>(1)
                                          |--STDEV<Float>(1)
                                          |---FLAG<Long >(1)
                        |----ITK<Long >(1)
                        |-RASTER<Long >(2)
                        |-STATUS<Long >(1)
                        |--GRPOS<Float>(2)
                        |--GRANG<Float>(2)
                        |--FPCUR<Float>(3)
                        |--FPPOS<Long >(1)
                        |--FPGAP<Float>(2)

OSIA > showstruct,aa
% Compiled module: SHOWSTRUCT.

(1)
   |----TYPE<Strng>(1)
   |-HISTORY<Strng>(1)
   |--HEADER<Strng>(1)
   |----DATA<Struc>(100)
                        |---WAVE<Float>(1)
                        |---FLUX<Float>(1)
                        |--STDEV<Float>(1)
                        |---TINT<Long >(1)
                        |----DET<Long >(1)
                        |----ITK<Long >(1)
```

```
|----UTK<Long >(1)
|---RPID<Byte >(2)
|--SPARE<Byte >(2)
|---LINE<Long >(1)
|---SDIR<Long >(1)
|---SCNT<Long >(1)
|-STATUS<Long >(1)
|---FLAG<Long >(1)
```

6.2 Structure description

All SWS Data structures consist of four main branches : type, history, header, data.

6.2.1 Type

Type is an ASCII expression which describes the type of the data product (e.g. SPD). This tag is used by some modules to identify the type of a structure (e.g. `is_xxx` , where xxx represents the data type).

6.2.2 History tag

This tag was planned to be filled with history information. But it is not, as yet, used.

6.2.3 Header

A very long string which contains the header information of the data product. As it is not advisable to apply pure IDL commands on this part of the structure, a number of modules have been written specifically to deal with it.

Some major modules to read and manipulate headers are :
`list_header` - to list the contents of the header.
`xmod_header` - to modify the contents of the header.
`read_fits_key` - to read certain header entries.
`write_fits_key` - to write certain header entries.
`delete_fitskey` - to delete a header keyword.
`compare_fitskey` - to compare header entries.

A typical example :

```
IA3 TEST> list_header,spd
% Compiled module: LIST_HEADER.
% Compiled module: LIST_HEAD_STR.
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                    8 / number of bits per data pixel
NAXIS   =                    0 / number of data axes
EXTEND  =                    T / FITS dataset may contain extensions
```

```
ORIGIN  = 'ESA     '           / European Space Agency
TELESCOP= 'ISO     '           / Infrared Space Observatory
INSTRUME= 'SWS     '           / Instrument used
COMMENT    SWS Standard Processed Data
FILENAME= 'SWSP02401183'       / File name
DATE    = '22/04/98'           / Creation date 98/112
FILEVERS= '1334    '           / Version ID (derived from creation date)
OLPVERS = 'OLP_632 '           / SOC OLP system version
CALGVERS= 'CALG_31 '           / SOC OLP CAL-G files version
USERNAME= 'PIPELINE'           /
INSTRA  =            316.75712 /Reference instrument J2000 right ascension (deg
INSTDEC =            +42.23587 /Reference instrument J2000 declination (deg)
INSTROLL=               231.99 /Reference instrument J2000 roll angle (deg)
CINSTRA =            316.75820 /Corrected reference instrument J2000 right asce
CINSTDEC=            +42.23583 /Corrected reference instrument J2000 declinatio
CINSTROL=               232.00 /Corrected reference instrument J2000 roll angle
OBJECT  = 'NGC7027 '           / Target ID as given by proposer
                        .
                        .
                        .
```

6.2.4 Data

The data part contains the actual product data. Here all product data are represented under similar tag name conventions as described in the Instrument Data Product Document.

6.3 Creating and manipulating IA^3 data structures

Data structures are automatically created by using the available read routines to read ISO SWS FITS data products (e.g. `read_fxxx` where xxx represent the data product, `readcal` to read calfibration files). The SWS interactive systems also provide tools to write data structures to FITS files (`write_fitstable`) (*see the HowTo-document* HowTo `read/write ISO/SWS data products and calibration files`).

To get an overview about the structures themselves, it is possible to use the IDL command `help, /struct`, but the SWS interactive systems contain also the program `showstruct` which displays the structure information in a more convenient form (see above).

If someone wants to create or manipulate an existing data structure the define_xxx (xxx : data product name) can be used e.g. `spd3=define_spd(data=spd1.data,header=spd2.header)`.

For display purposes, it is also possible to choose between pure IDL commands (e.g. `plot,aa.data.wave,aa.data.flux` and special tools provided by the ISO-SWS software (e.g. `plotaar,aa`)

Of course it is possible to apply pure IDL functions on the data structures.
E.g. `aar.data.flux = aar.data.flux + 22.`

7 HowTo understand and manipulate flags / statuswords

ISO SWS standard data products (ERD/SPD/AAR) contain flag and statuswords to describe the status of the instrument and pass additional information. These words are represented as Bit code. After reading the ISO SWS standard data products (*see* HowTo read/write ISOSWS data structures) into IDL structures (*see* HowTo understand and manipulate ISOSWS data structures) it is possible to decode or manipulate the flag/statuswords. Flags and statuswords contain necessary information on the origin and instrument conditions for each and every datapoint. The first two sections of this HowTo describe the content of flags and status words. While their manipulation is discussed the subsequent two sections.

7.1 Status words

Status words reflect the status of the instrument. They are given per reset interval. Status words are used in many processing steps by many modules to identify the status of the instrument.

The aperture bits indicate the shutter position closed, one, two or three (table 1).

Table 1: Status word Bit codes: Aperture

Description	Bit code	number
aperture 1	2^0	1
aperture 2	2^1	2
aperture 3	$2^0 + 2^1$	3

Reset pulses are particular of interest if different reset intervals are commanded for the short wavelength (SW) and long wavelength (LW) bands (table 2).

Table 2: Status word Bit codes: Reset

reset band 1	2^2	4
reset band 2	2^3	8
reset all bands	$2^2 + 2^3$	12

The settings of internal calibrators are reflected in the calibrator status bits (table 3).

Table 3: Status word Bit codes: Calibrator status

diffuse cal normal	2^4	16
diffuse cal high	$2^4 + 2^5$	48
FP check normal	2^6	64
FP check high	$2^6 + 2^7$	192
flusher normal	2^8	256
flusher high	$2^8 + 2^9$	768
grating check normal	2^10	1024
grating check high	$2^10 + 2^11$	3072

Which detector bands are requested for the actual observation is information that is directly passed from the uplink to the downlink, so called TDATDA. They are not considered in any interactive tool (table 4)

The status of the Fabry-Perots (which FP is active and running etc.) is indicated in the FP bits (table 5).

Table 4: Status word Bit codes: Requested bands

BAND1 requested	2^13	8192
BAND2 requested	2^14	16384
BAND3 requested	2^15	32768
BAND4 requested	2^16	65536
BAND5 requested	2^17	131072
BAND6 requested	2^18	262144

Table 5: Status word Bit codes: Reset

FP 2 active	2^12	4096
FP execute flag	2^19	524288
FP run flag	2^20	1048576

Scans where the internal calibrator is switched on (photometric check), a reference or dark current scan is done or the low resolution mode is used are marked by the scan type bits (table 6)

Table 6: Status word Bit codes: Scan type

low resolution scan	2^21	2097152
reference scan	2^22	4194304
photometric check	2^23	8388608
defined dark measurement	2^24	16777216

The scan direction bit is set in the case of up scans (table 7)

Table 7: Status word Bit codes: Run flag and Scan direction

SW grating run flag	2^25	33554432
LW grating run flag	2^26	67108864
SW scan direction	2^27	134217728
LW scan direction	2^28	268435456

7.2 Flag words

The flag words contain additional information which might be different for different detectors within one reset interval. They are given per reset interval for every single detector.

The flag word contains the number of glitches within a reset interval. Glitch affected data are doubtful and have to be treated with care. Also data which are totally or partly saturated are of doubtful character. Data marked as 'no data' are not usable (table 8).

Table 8: Flag word: doubtful data

Description	Bit code	number
one glitch	2^0	1
two glitches	2^1	2
more glitches	$2^0 + 2^1$	3
partially out of limit	2^2	4
all points out of limit	2^3	8
no data	2^4	16

The observed wavelength order is indicated by the flag word as well (table 9).

Table 9: Flag word: grating order

Description	Bit code	number
grating order one	2^5	32
grating order two	2^6	64
grating order three	$2^5 + 2^6$	96
grating order four	2^7	128
multiple grating orders	$2^5 + 2^6 + 2^7$	224

Certain interactive tools (`mask_inter`) set the mask flag to signal data which should be ignored for further data processing(table 11).

7.3 Decoding

Knowing the BIT code it is easy to decode and manipulate status and flag words using simple IDL AND/OR operations. Nevertheless SWS interactive systems provide the user with a number of tools especially adapted to deal with SWS status and flag words.

Table 10: Flag word: gain settings

Description	Bit code	number
gain one	2^9	512
gain two	2^10	1024
gain three	2^9 + 2^10	1536

Table 11: Flag word: Masking

Description	Bit code	number
mask flag	2^30	1073741824

7.3.1 Status words

`test_status`
is one of the basic modules to treat status words. It is possible to check status word bit codes on certain status values. It is very effective to use it together with the `select` tool. E.g. to select the first dark current scan block
`spd2=select(spd,test_status(spd,/dark),block=1)`

`decode_status`
is decoding the status word and returning an ASCII output.

```
OSIA> print,decode_status(spd.data(100:101).status)
% Compiled module: DECODE_STATUS.
% Compiled module: TEST_STATUS.
% Compiled module: MAKE_STATUS.
% Compiled module: IS_SPD.
% Compiled module: IS_ERD.
% Compiled module: IS_AAR.
APER =1     SWRESET     LWRESET     SWRUN       LWRUN       SWDOWN     LWDOWN
APER =1     SWRESET     LWRESET     SWRUN       LWRUN       SWDOWN     LWDOWN
```

`plot_status` & `plot_spd/plot_aar`
The different plot routines (e.g. `plotspd` or `plot_status`) indicate the status of the observations. For `plot_spd` amd `plot_aar` this is possible only when the keyword /STATUS is given.

7.3.2 Flag words

`test_flag`
is one of the basic modules to treat flag words. It is possible to check flag word bit codes on certain flag values. It is very effective to use it together with the `select` tool. E.g. to select all data with more than two glitches

```
aar2=select(aar,test_flag(aar,glitch='M')).
```

For SPD (an explicit flag must be selected)

```
spd2=test_flag(spd.data.det[0].flag,glitch='M').
```

`decode_flag`

decodes the flag word and returns an ASCII output.

```
OSIA> print,decode_flag(spd.data(100:101).det(30).flag)
% Compiled module: DECODE_FLAG.
% Compiled module: TEST_FLAG.
% Compiled module: MAKE_FLAG.
ORDER : 2, GLITCH : 0, OOL : NONE , ORDER : 2, GLITCH : 0, OOL : NONE
```

plot_flag
Also the different plot routines (e.g. plotaar) indicate different flag settings if the keyword /FLAG has been given.
plot_flag & plot_spd/plot_aar
The different plot routines (e.g. plotspd/pltoaar or plot_status) indicate the status of the observations. For plot_spd and plot_aar this is possible only when the keyword /FLAG is given.

7.4 Manipulate

Sometimes it is useful to manipulate flag or status words. This is possible using simple IDL AND/OR commands once one knows the bit code. The SWS interactive systems also provide a number of tools for flag and status word manipulation.

Tools like make_flag or make_status and mod_flag or mod_status might be used to create and manipulate flag and status words directly. But also a number of interactive tools manipulate the flag words internally (e.g. mask_inter is setting the MASK BIT in the flag words) or fluxcon or dark set NO_DATA flags in the case of problems or to mask out certain data.

8 HowTo link ISAP to you OSIA session

8.1 Introduction

This recipe assumes that you have already installed ISAP on your machine
(*see the following URL* http://www.ipac.caltech.edu/iso/isap/isap.html).

ISAP v2.0 (and higher versions) provides a routine LINK_ISAP, which allows one to link ISAP to any IDL session (e.g. OSIA). This interface is also useful for handling non-ISO spectra (see recipe for isap_yaaar in ISAP's doc/recipes/ directory).

8.2 Procedure

If you want to link ISAP within an existing IDL session simply copy the file link_isap.pro (in the pro/mpe/ directory) to your current directory (or any directory in your IDL path), and edit the line which defines the path to the top ISAP directory. Then execute the file like

```
IDL> @link_isap
```

You can also copy this line to your IDL startup file. This will give you full command line access to all ISAP routines. To start the ISAP GUI simply type

```
IDL>start_gui
```

If you have been working on an AAR in your IDL session, e.g. if you have produced it within the OSIA environment, you can pass this aar directly to the GUI:

```
OSIA> ...
OSIA> NGC1068 = aar
OSIA> start_gui,aar_in=NGC1068
```

In this example the name of the aar inside the GUI will be "NGC1068". Vice versa, the GUI - on exit - passes an AAR back to the command line. To retrieve this aar type

```
IDL> aar_out=gui_output()
```

after you have exited the GUI.

8.3 Fixes

For historical reasons there is a slight inconsistency between SWS AARs in OSIA and ISAP: the two data tags SPARE and RPID are byte arrays in OSIA, but integers in ISAP. Should this ever cause problems in your application you can use the ISAP routine aar_fix to transform the AAR from one representation to the other:

```
OSIA> isap_aar = aar_fix(osia_aar) ; (the mode 'ia2isap' is default)
OSIA> ...
OSIA> new_osia_aar = aar_fix(isap_aar,mode='isap2ia')
```

Part II

OSIA Data Processing

9 HowTo get a quicklook at your observation

This HowTo gives an overview about tools which are suited to obtain a quick look at your observations.

9.1 Getting help with routines

The routines described here are presented with a few examples which cover only a small fraction of the routine's capabilities. We strongly suggest that when reviewing the routines in this HowTo (and others) that you also review the keywords present in each of the routines. This can be accomplished by using the `widget_olh` command at the OSIA prompt or the `osia_man` in your normal shell prompt. `widget_olh` will pull up manual pages for all the routines in the OSIA release as well as pages for the calibration files and the general ASTRON library. `osia_man` runs outside OSIA and allows one to browse for specific OSIA commands and to perform keyword searches.

9.2 Standard Plots and Interactive tools

Standard plot routines like `ploterd,plotspd,plotaar` give a first impression about the actual observation in certain processing stages. Keywords like `flag` and `status` give even more information.

The `plotaar` routine plots all or part of an AAR. These plots are ordered in wavelength (default setting):

`plotaar,aar` ;which, by default, plots everything within the AAR.
`plotaar,aar,band=[1,2]` ;plots only data for AOT bands 1a through 2c.
`plotaar,aar,band='2b'` ;plots only data within AOT band 2b.
`plotaar,aar,band='2b',/flag,/status` ;plots AOT band 2b data and decodes the `flag` and `status` information given below the AAR plot.

The `plotspd` routine is similar to plotaar but plots all or part of an SPD. These plots are ordered in time during the observation (default setting):

`plotspd,spd` ;plots the first detector in the SPD.
`plotspd,spd,band=1` ;plots all the band 1 detectors on the same plot.
`plotspd,spd,det=7` ;plots all the detector 7 data.

The `plotaar` and `plotspd` routines have many available keywords to help the user explore the data and control the way the plot looks.

As you become better acquainted with SWS and its data products, you may wish to look at the very raw data in the ERD. The `ploterd` routine will help with this [(*see the dc-document* `ploterd.pro`) also see How to process SWS data from ERD to SPD].

It is also possible to use interactive tools such as `mask_inter` to get a first overview about an observation on a scan by scan basis.

9.3 Scan Tables

Scan tables give an overview of observations at the SPD level. Certain interactive processing steps require scan tables as input parameters as well.

```
sctbl=make_sctbl(spd)
```

(*see the dc-document* `make_sctbl.pro`) creates a scan table structure (sctbl) out of the given SPD. This structure is very complex and therefore the SWS interactive systems provides a tool to visualize the scan table.

```
print_sctbl,sctbl
```

(*see the dc-document* `list_sctbl.pro`) displays the structure items in an convenient form on the screen or writes the output to an ASCII file.

A scan table contains the following information :

Extracts from the header :

scan table	:	observation number
created:	:	scan table creation date
Proposer	:	Proposer ID
Proposal	:	Proposal ID
ISO	:	Instrument
AOT	:	AOT number
Target	:	Target identification as given by the proposer
Obs_ID	:	Observation ID
Start_TDT	:	Start TDT number
End_TDT	:	End TDT number
RA [hms]	:	Right Ascension J2000 [hms]
DEC [dms]	:	Declination J2000 [dms]
ROLL [deg]	:	Roll Angle J2000 [deg]
CRA [hms]	:	Corrected Right Ascension J2000 [hms]
CDEC [dms]	:	Corrected Declination J2000 [dms]
CROL [deg]	:	Corrected Roll Angle J2000 [deg]
V	:	ISO velocity towards target [km/s]

Observation blocks (for every scan/dark current measurement or photometric check) :

nr	:	Block number
AOT	:	AOT number
scan_type	:	Scan type (Dark current, gratcal=low, upscan, downscan...)
start	:	Start reset interval
return	:	Scan return point
end	:	End reset interval
step	:	Step size
grpos_min	:	Minimum grating position
grpos_max	:	Maximum grating position
a	:	Aperture
r	:	Reset interval
band	:	Detector band
or	:	Order
gain	:	Gain setting
wave_min	:	Minimum wavelength
wave_max	:	Maximum wavelength

9.4 Quicklook

`quicklook` is a tool that creates an IDL script file. Executing this script allows one to easily preview/print lines/ranges of AOT SWS01, SWS02 or SWS06 observations. Additional possible parameters are explained in the program documentation (*see the dc-document* `quicklook.pro`). The parameters point, filled and jitter should be used to characterise the observed source.

For example :

```
spd = read_fspd(spdfilename)
sctbl = make_sctbl(sctbl)
quicklook, sctbl, file='quicklook.pro'
```

produces the following quicklook.pro which may be executed at the OSIA prompt as `@quicklook`. The associated auto analysis product is named AAR for this purpose (as one can see from the code) :

```
  if not is_aar(aar) then print,'==> aar not found !'
  xt='Wavelength [!4l!3m]'
  yt='Flux [Jy]'
; target = NGC7027
; ------------------------------------------------------------------------
  !P.MULTI=[0,1,4] ; n_page = 2
; ------------------------------------------------------------------------
; SW1/AP=1 # 0: 1 scans  tr= 2  gain=16  R=1557  w= 2.7051  w0= 2.7051  RANGE_1
  line=aarfilt(aar,[ 2.360, 3.050], 1557., kappa= 3.0, /no)
  fdif=(MAX(line.data.flux)-MIN(line.data.flux))>0.5
  fmin=(MIN(line.data.flux)-0.1*fdif)<0.0
  fmax=(MAX(line.data.flux)+0.1*fdif)>0.0
  plot,line.data.wave,line.data.flux,title='NGC7027 : RANGE_1', $
       xr=[ 2.360, 3.050],/xs,xtit=xt,yr=[fmin,fmax],ytit=yt
; ------------------------------------------------------------------------
                                *
                                *
```

```
                                *
; ------------------------------------------------------------------------
; LW2/AP=3 # 7: 1 scans  tr= 2  gain=1   R=1317  w=37.0726  w0=37.0726  RANGE_8
  line=aarfilt(aar,[28.751,45.394], 1318., kappa= 3.0, /no)
  fdif=(MAX(line.data.flux)-MIN(line.data.flux))>0.5
  fmin=(MIN(line.data.flux)-0.1*fdif)<0.0
  fmax=(MAX(line.data.flux)+0.1*fdif)>0.0
  plot,line.data.wave,line.data.flux,title='NGC7027 : RANGE_8', $
       xr=[28.751,45.394],/xs,xtit=xt,yr=[fmin,fmax],ytit=yt
; ------------------------------------------------------------------------
  !P.MULTI=[0,0,0]
; ------------------------------------------------------------------------
```

9.5 Plot_phot

`plot_phot` visualizes the photometric checks in an SPD. It is possible to select by detector, photometric check number (if > 0), over a wavelength range, in single plots for comparison, overplotted in one figure for comparison etc.. This tool enables users to immediately recognise problems in photometric calibration scans. For example whether glitches cause outliers which are not smoothed by the three point median filter of `fluxcon`, or if a photometric check differs significant from the photometrc checks of the other detectors.
E.g. `plot_phot,spd,sctbl,1,30,/all`

9.6 Quality Check

A true quality check on your data can only be assesed by you. However, we can point to places where, during the processing, problems might arise. The number of problems possible with SWS is not really large and usually there are procedures for correcting them. Below is a list of things to check out in your SWS data.

To maximise the use of the quick-look, we recommend that you faimiliarise yourself with the reduction processing steps by referring to the ISO SWS Handbook (*see the following URL* `http://www.iso.vilspa.esa.es/manuals/HANDBOOK/sw`
Also HowTo process SWS data from ERD to SPD and HowTo process SWS data from SPD to AAR should be read before analysing the results of your quick-look.

- Reasonable Dark Currents?
  The standard processing does a pretty good job in determining the dark currents, but nature can still get in the way and cause the detectors to do funny things.

  The dark currect subtraction is easily checked using the `plotspd` command. First produce a dark subtracted SPD in a different variable name:

  ```
  spd_dark=dark(spd)
  ```

  then plot both SPDs with `plotspd`. If any one detector is very different from the rest, it may indicate a dark current problem. In this case try the `dark_inter` routine to fix it How to perform dark subtraction with DARK_INTER.

- Reasonable Photometric Gains?
  The procedure `plot_phot` lets you see individual photometric checks. This routine is useful for AOT7 data since the photometric gains can be strongly influenced by uncorrected glitches.

  However, in OLP10, only one gain is calculated per detector band. So from a dark subtracted SPD you can do the following to find the photometric gain of your observation:

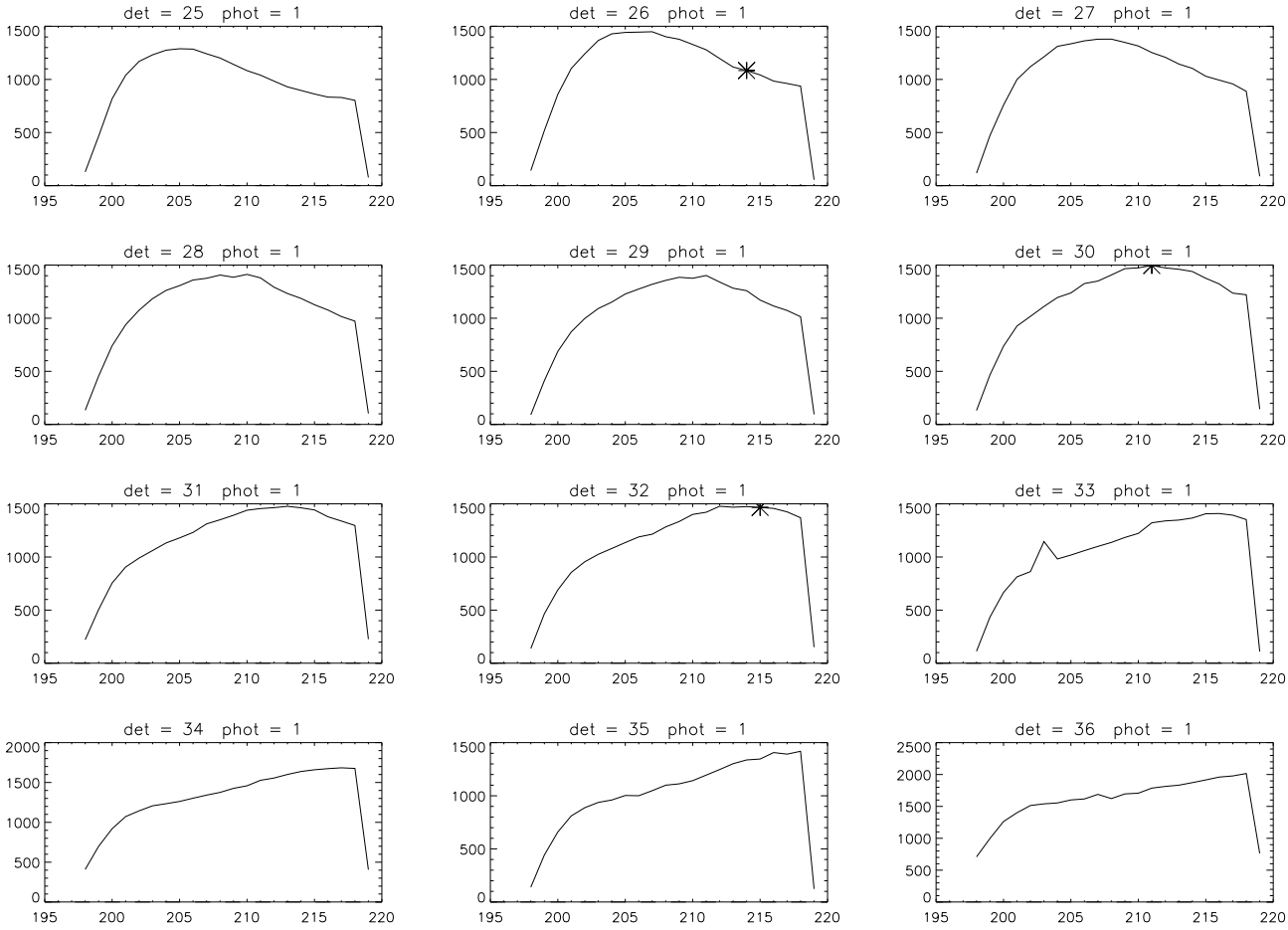
det = 25 phot = 1
det = 26 phot = 1
det = 27 phot = 1
det = 28 phot = 1
det = 29 phot = 1
det = 30 phot = 1
det = 31 phot = 1
det = 32 phot = 1
det = 33 phot = 1
det = 34 phot = 1
det = 35 phot = 1
det = 36 phot = 1

Figure 3: Example : PLOT_PHOT output

`gain=get_phot_gain(get_phot_check(spd))` then `print,gain([0,12,24,36])` which will print the photometric gains calculated for data in bands 1, 2, 3 and 4. If you have an AOT 7 then only detectors 49 and 51 have useful gain information for bands 5 and 6.

- Is the pointing OK?
  see How to correct mispointed observations
- Fringes in the Spectrum?
  see How to minimize fringe residuals in SWS data

10 HowTo process SWS data from ERD to SPD

ISO SWS data are processed by the ISO SWS pipeline software and can be retreived from the ISO Data Archive (IDA). Within OSIA it is possible for the user to derive the SPD data from ERD enabling the user to check or refine the pipeline processing (Wieprecht [16] gives a description of the SWS software systems).

This HowTo describes how to process SWS data from the set of Edited Raw Data (ERD) to Standard Product Data (SPD) as it is done in the pipeline. This means removing all artifacts from the detectors and the electronics seen in the integrated 24 Hz ramps, converting the 24Hz detector readouts (bits) into slopes [μV/sec] and assigning, where possible, wavelengths. In figure 4 a simplified scheme of the SWS read-out electronics is given and in Lahuis [5] the read-out process is described in more detail.

10.1 Accessing data

First, the ERD data has to be read in and, if necessary, calibration files selected.

```
erd = read_ferd( '<obsnumber>', path='<path>' )
```

This reads the SWER, SSTA, GEHK and EOHI telemetry data to create the basic IDL ERD structure (these files should be present to correctly read in the ERD). When present the IRPH and SWSP standard FITS files are read in and used to update the ERD header. The IRPH file is used to extract the reconstructed pointing information and the SWSP file is read to retain the ISO velocity information which is not present in the basic ERD files. When the velocity information is not present, the SPD wavelengths *cannot be velocity corrected in the DAAR processing* (see HowTo process SWS data from SPD to AAR).

10.1.1 Select calibration files

If you want to use special calibration files it is necessary to select these files before processing your data. The selected calfiles can be defined globally via `CAL_SELECT` or read into OSIA as an `IDL` structure with `READCAL`. The calibration structure can then be modified in OSIA and entered on the command line to one of the routines.

10.2 Derive Standard Processed Data - `DSPD`

To run the complete pipeline processing from ERD to SPD (SWSP file) use:

```
spd = dspd( erd )
```

The `DSPD` processing is split up into separate parts each of which can be called separately. The following processing steps explained sequentially in the following sections will give exactly the same result as the above call. In this way intermediate products can be checked at each level of processing.

The following sections give a short description of all separate `DSPD` processing steps. The title of each section is intended to be descriptive and it also gives the corresponding `OSIA` routine used for this step and the most relevant calibration files involved (see section 10.3 for a short description). In all cases the SWS handbook (*see the following URL* `http://www.iso.vilspa.esa.es/manuals/HANDBOOK/sws_hb/`) and the accompanying dc documents of the routines and the calibration files (`widget_olh` or `osia_man`) can be consulted for more detail.

Figure 4 gives a schematic representation of the SWS read-out electronics which is the origin of the artifacts described here. A more detailed description is given in Lahuis [5] and in the SWS handbook.

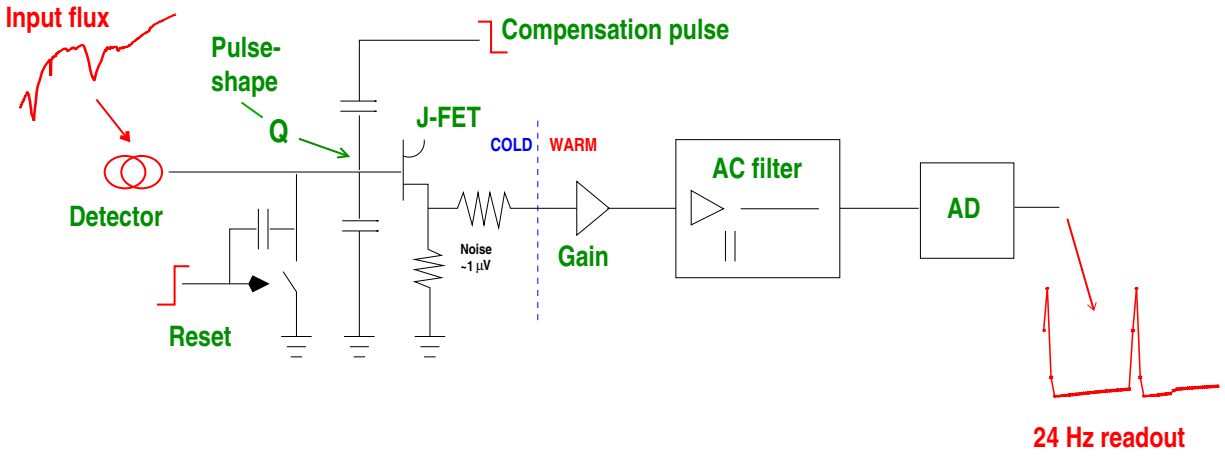

Figure 4: A schematic representation of the SWS read-out electronics indicating the major components which affect the integrated detector signal. In section 10.2, the implemented corrections for these are described. A more detailed description of all the known artifacts is given in Lahuis [5].

10.2.1 Valid data range - `DRANGE (CAL04)`

```
erd = drange( erd )
```

This step has to be done before any other data processing. At the bit level, invalid data (e.g. out of limit data) are recognised using calibration file 04 which defines the lowest and highest bit values at which a sample becomes invalid. The invalid values are masked by -Inf (violating the lower limit) and +Inf (violation of the upper limit). Inf is defined with the IDL system variable `!VALUES.F_INFINITY`.

10.2.2 Electronic zero level - `MIDBIT (CAL02A)`

```
erd = midbit( erd )
```

The detector readout voltages of between -10 and + 10V are converted into a bit number ranging from 0 to 4095. The MIDBIT is the bit value that corresponds to approximately zero Volts. It is defined in the calibration file 02 as 2047.5 for all detectors.

The value of MIDBIT is not excact and may depend on reset time or gain. However, this does not matter too much. An incorrect value for MIDBIT will show up as an added component to the signal and only gives an extra uncertainty in the absolute zero level. This is not harmful since this is corrected for through the dark current measurements in the observation.

10.2.3 High-pass filter - `ACCOR (CAL02,03)`

```
erd = accor( erd )
```

The amplifier chain in the detector electronics contains a high pass filter. For a constant voltage gradient input (from a constant signal level falling on the detector) the response of the amplifier asymptotically approaches a constant value.

In order to later reconstruct the true photo-current, it is necessary to linearise the integration ramps. Using the RC time-constant (stored within calibration file 02 for each detector) it is possible to derive the input voltages. The linearisation is applied for all samples excluding the first three or the number of samples defined in Calfile 03 if this is less than three.

The high pass filter is corrected in the pipeline using the following approximation:

$$D_i = d_i + \sum [\frac{d_j + d_{j-1}}{\frac{1}{2}\tau}]$$

where $\tau = 24.0/(2 \cdot \pi \cdot \tau_f)$ with τ_f the RC time-constant from calfile 02 in terms of frequency.

10.2.4 Reset pulse after effect - PULSE_SHAPE (CAL03,05)

```
erd = pulse_shape( erd [, /plot] )
```

In PULSE_SHAPE, a correction is made for the after-effect of the reset and compensation pulse applied after each reset interval. After the application of these a charge can be left on the input to the JFET's. This charge decays over the input capacitance of the JFET's and is seen as an exponentially decaying signal added to the input signal from the detectors. The curve has typical decay times of a few seconds and an amplitude of up to 50 μV. The pulse-shape effect can change from observation to observation and is in cases also seen to change within an observation. The correction is therefore derived from the ERD data itself and is allowed to change in time.

The pulse-shape correction is derived in the following manner:

- To all integration ramps slopes are fit and pulse-shapes are derived from the slope residuals. The first samples as defined in Calfile 03 are excluded in the slope and pulse-shape fitting.
  - Slopes are fit (see section 10.2.6) to all ramps and the residual curves (in μV) are returned.
  - To the residual curves an exponential plus slope is fitted.

    $$p = A * e^{-\tau t} + B + C * t$$

    where B and C can be expressed in terms of A and τ. This limits the number of free parameters to only two, A and τ, the amplitude and decay time of the exponential pulse-shape. The pulseshape fitting is done in tow steps:
    - An average residual curve is calculated and a pulseshape fit is made to derive the exponential decay time used for the whole observation.
    - To all residual curves pulseshape fits are made with the fixed decay time derived from the average residual curve. From this the pulse-shape amplitudes are derived.
  - The time-varying pulse shape amplitude is smoothed to reduce the noise.
- The exponential pulse-shapes are subtracted from the complete integration ramps.

The derived and subtracted pulse-shapes can be checked by running pulse_shape with the /plot option.

A more extensive description of the pulse shapes and the implementation in the pipeline is given in Lahuis [8].

10.2.5 Detector cross-talk - CROSSCOR (CAL01)

```
erd = crosscor( erd )
```

Due to parasitic capacitance between neighbouring detectors any signal in one grating detector will 'leak' to other detectors (primarily the adjacent ones) in the same array. Calibration file 01 contains a matrix which is applied to correct for this effect.

The crosstalk matrix is applied over all detectors for each 24 Hz read-out in the integration reset. If during a read-out one detector is out of limit (see section 10.2.1) then all detectors are flagged as out of limit for that read-out sample.

10.2.6 Slope calculation - SLOPE (CAL03,05,06)

```
spd = slope( erd, [gll] )
```

In the slope calculation a number of intermediate steps are involved. These steps do not need to be called separately but are run from within SLOPE. The whole process is done as follows:

- For all integration ramps the first samples which are directly affected by the reset are skipped. Calibration file 03 defines the number of samples which are skipped. Currently this is defined as six samples.
- The data are analysed for glitches.
  - the derivative of the integration ramp is calculated
  - for low resolution AOT-1 data the source signal component is subtracted from the derivative (see Lahuis [9])
  - glitch detection is performed upon the ramp derivative
- The detected glitch data is saved in the Glitch structure (gll - which is optional). In OLP the glitch data is stored in the SWGH file.
- Slopes are fit to the (glitched) integration ramps.
  - for low resolution AOT-1 data the slopes are first corrected for glitches using the glitch height estimated in the glitch detection
  - for all other data a simultaneous fit to the slopes and glitches is made
- The slopes are converted into μV/s.
- The flag and status words are updated.
- The SPD structure is filled and returned.

10.2.7 Assign wavelengths - WAVE (CAL12,16(a,b,c,et),18,23,24)

```
spd = wave( spd [, erd] )
```

After computing the FP gaps and grating angles the wavelengths are calculated for each unique grating position, order, aperture and detector setting.

For AOT1, AOT7 and AOT99 CUS observations the ERD structure is required to do a proper wavelength calibration.

10.2.8 Write the Standard Product Data

At the end the SPD can be saved as a standard FITS file with write_fitstable.

```
write_fitstable, spd, 'MySPD.fits'
write_fitstable, gll, 'MyGLL.fits'
```

It is also possible to save the SPD as an IDL saveset. The SPD can then be further procssed with the DAAR software (*see the HowTo-document* HowTo process SWS data from SPD to AAR).

10.3 Calfiles used in DSPD

Below a one-line description of all calfiles used in the DSPD processing is given. For more details about the contents and the data used in the derivation of the individual files the OSIA dc documents should be consulted.

DSPD CALFILES

CAL01 electronic cross-talk matrix

CAL02 RC timeconstants

CAL02a zero Volt point in bitrange

CAL03 reset cutout length

CAL04 amplifier limits

CAL05 switchable gain values

CAL06 glitch reject levels

CAL12 position - gap relation of scanning FP's

CAL16a aperture offset angles

CAL16b detector element offset angles

CAL16c grating constants

CAL16et grating polynomial coefficients

CAL18 wavelength dependent FP gaps

CAL23 wavelength limits of detector bands

CAL24 wavelength limits of apertures

11 HowTo process SWS data from SPD to AAR

ISO SWS data are processed by the ISO SWS pipeline software and retrieved from the ISO Data Archive (IDA). Within OSIA it is possible for the user to derive the AAR data from SPD enabling the user to check or refine the pipeline processing (Wieprecht [16] gives a description of the SWS software systems).

In the SWS handbook a detailed description of the pipeline processing is given. In this HowTo we will therefore concentrate mostly on those routines in OSIA which replace or complement some of these steps.

11.1 Introduction

The major steps in the DAAR processing are the dark current subtraction, the application of the RSRF, the conversion from μV/s to Jy and the ISO velocity correction. All of these corrections can be applied in OSIA as it is done in the pipeline. For a few steps, interactive OSIA routines are also available.

For specific details concerning the use of the routines the OSIA dc-documents should be consulted.

In Appendix 11.7 the most common error messages issued by the pipeline modules are listed.

11.1.1 Standard pipeline

The standard pipeline processing of the SPD data to an AAR is described in detail in chapter 6.3 of the ISO-SWS Handbook (*see the following URL* http://www.iso.vilspa.esa.es/manuals/HANDBOOK/sws_hb). The different steps are addressed in the order of processing in the standard pipeline and the OSIA programme names are listed. In chapter 6.4 the error propagation is described.

11.2 OSIA DAAR processing steps

In OSIA a few pipeline steps can be replaced or followed-up by specialised OSIA routines. These are the dark current subtraction and the RSRF correction.

The available OSIA routines for specific DAAR processing steps (section 11.3) are described in the following. First, in Section 47 to 11.2.5 we present a few general utilities which are deemed useful for DAAR processing. Besides these there are a large number of other routines in OSIA which allow the user to access, examine or manipulate SPD data.

There is a mix of command-line routines and interactive routines. Some of these routines commensurate and it may depend on the particular use or preferences of the user which will work best. All have disadvantages and advantages and it will depend on the type of processing done which will be most suitable. A clear advantage of command line tools is that they can easily be used in scripts allowing straightforward reproducibility. Sometimes the interactive tools are convenient to inspect the data and locate problems. A solution may then be found which can then be implemented using one of the command line tools in a script.

In all cases the user should first read the OSIA dc-documents and take account of all the options available in the routine. The commands shown here are mere examples (with sometimes a few extra options given in between square brackets). Often many more options are possible. These are described in the OSIA dc-documents.

In the following we describe simply the sequential order of commands that should be followed to correctly reduce your basic SPD to an AAR. Where possible we list alternative routines that can be used to replace automated modules. Please refer to the ISO-SWS handbook, further information given below in this HowTo and dc-documents for detailed descriptions of the commands, syntax, usage and options.

11.2.1 Order of commands and basic syntax

Step	Action	Available modules
1	Read-in the SPD:	READ_FSPD
2	Make Scantable:	MAKE_SCTBL
3	Reduce Memory Effects:	ANTIMEM
4	Dark Current Subtraction:	DARK or DYNADARK (NB for band 2 DARK uses DYNADARK) or DARK_INTER (and MASK_INTER, PATCH_SPD, UPDOWN_INTER(only AOT2s))
5	Relative Spectral Response:	RESPCAL or RESP_INTER (and FRINGES)
6	Absoulte Flux Calibration:	FLUXCON
7	Velocity Correction:	VELCOR
8	Extract AAR Product:	EXTRACT_AAR (and UPDATE_LINECOUNT)
9	Write AAR FITS File:	WRITE_FITSTABLE

[Commands in round brackets **may** also be implemented after one of the preceding commands.]

11.2.2 Select calibration files

If you want to use any particular calibration files it is necessary to select these files *before* processing your data. The selected calfiles can be defined globally via CAL_SELECT or read into OSIA as an IDL structure with READCAL. The calibration structure can then be modified in OSIA and entered on the command line to one of the routines.

11.2.3 Creating the SPD IDL structure

To read in the offline pipeline processed SPD the following command may be used

```
spd=read_fspd('filename')
```

(*see the dc-document* read_fspd.pro) where filename is the name of the ISO data product SWSP FITS file. If the SPD structure was created within OSIA from the raw data products (*see the HowTo-document* HowTo process SWS data from ERD to SPD) then the SPD may be read in as given above with filename being the name of the SPD FITS file created by the user or, if continuing processing in one IDL session or if restoring an IDL save file, then reading in the FITS file is not required.

11.2.4 SPD scantable

The SPD scantable gives an overview of the SPD contents. Besides being informative of nature it is also required input to some of the interactive routines.

```
sctbl = make_sctbl( spd )
```

The scantable can be printed to screen or file with print_sctbl.

```
print_sctbl, sctbl [, filename=filename ...  ]
```

For SWS06/07 observations, improved or proper operations of some reduction tools can be obtained by converting the scan table into a pseudo-SWS02 scan table where the separate SWS06 up- and down-scans are converted to a SWS02-type updown scan. This conversion is performed with the command convert_sctbl.

```
sctbl = convert_sctbl( sctbl , filename='mask_6to2.run' )
@mask_6to2.run
```

The script created by `convert_sctbl`, called mask_6to2.run in this example, modifies the contents of the SPD flags. Its assumes the IDL variable name is SPD.

11.2.5 SPD plotting

An essential feature in the processing is of course the data visualisation. The main routine for this is `plotspd`.

```
plotspd, spd [, det=det, /status, /flag ....  ]
```

It plots the contents of the SPD structure. Many different ways of plotting the SPD data are available. It is possible to plot e.g. flux against time, wavelength or grating position, plot the stdev, the grpos. It can be very helpful in locating problems or checking the results of a processing step. With the `/status` and `/flag` options the status of the instrument and of an individual detector can be plotted. In HowTo understand and manipulate flags / statuswords the meaning of these are explained.

The status and flag can also be plotted with a few stand alone routines.

```
plflag, spd, <det> [, xrange=xrange ]
plot_fullflag, spd [, band=band ...  ]
```

Plot the bits of the the data flag and their meaning.

```
plstat, spd [, xrange=xrange ]
plot_fullstatus, spd [, ...  ]
```

Plot the bits of the the status flag and their meaning.

11.3 OSIA DAAR routines

The following routines either replace one of the pipeline processing steps or can be used in combination with one of the existing pipeline routines. In all cases the pipeline routine in question is listed in brackets in the section header.

11.3.1 Reproducing pipeline reduction in OSIA

The pipeline AAR can be reproduced by using the command

```
aar=daar(spd)
```

(*see the dc-document* `daar.pro`). The steps involved are explained in detail in the ISO SWS Handbook (*see the following URL* `http://www.iso.vilspa.esa.es/manuals/HANDBOOK/sws_hb`).

The individual steps involved in this reduction stage are listed in order in section 11.2.1. The subsequent sections detail some of these steps.

11.3.2 Transient Correction

antimem ideally removes transient effects i.e. detector memory effects. Please refer to Section 7.3.2 of the ISO-SWS handbook for more details and the dc-document antimem.pro for syntax and usage.

11.3.3 Dark Current Subtraction

Please note: When applying ANY of the dark current subtraction routines in OSIA a floating-point exception error may occur. This is simply a warning and is normal for these routines, please do not be alarmed by this message and continue with the reduction as normal.

11.3.3.1 dark The pipeline dark current subtraction can be replicated by using the procedure dark (please refer to the dc-document dark.pro for syntax and usage). This sophisticated routine automatically determines and subtracts dark currents from the data. Please refer to Section 7.3.3 of the ISO SWS Handbook for more information.

11.3.3.2 dynadark (dark) In the pipeline (in OSIA this means in DAAR and DARK) DYNADARK is used for the transient correction for the band 2 Si:Ga detectors with the Fouks memory model (see Kester[4]).

dynadark can however also be used for the dark current subtraction in any of the other bands and thereby replacing dark. For this purpose a model 'none' exists. This is the default model for all other bands. The author has found the results to be consistent with the standard dark current subtraction and has achieved good results in some difficult cases where the normal dark fails.

```
spd = dynadark( spd, detlist=indgen(12)+1 )
```

For the band 4 Ge:Be detectors, it is also possible to use the Fouks model. Although this is not the proper physical model for these detectors, improvements can be achieved when applying it to these detectors. In that case it is often best to leave all model parameters free and to make sure to check the results. In dynadark the plot and verbose options are present to help in checking the fit results.

```
spd = dynadark( spd, detlist=indgen(12)+37, model='fouks', fixed=-1, /plot,
/verbose )
```

Also for band 2 data with very high signals (¿few $10^3 \mu$V/s) the dynadark results can sometimes be improved by letting dynadark fit all parameters.

```
spd = dyndark( spd, detlist=indgen(12)+13, fixed=-1, /plot, /verbose )
```

11.3.3.3 dark_inter (dark) dark_inter offers a highly interactive dark current subtraction over which the user has direct visual control. As the quality of the result is driven by visual inspection and, if necessary, detailed dark correction for individual scans, using this tool can be rather time-consuming.

In most cases the optimal results will be similar those of the standard dark routine. However, looking carefully at the data in dark_inter can help the user to become familiar with the dataset and to recognise problems such as single/multi detector jumps, strong cosmic ray hits with long aftermath and the influence of memory effects at an early stage of data reduction.

It should be noted that for band 2 `dark_inter` is not compatible with the pipeline implementation. In the pipeline (and in the OSIA `dark` routine) a dynamic dark current subtraction using the Fouks memory model is used (see Kester [4] for details).

A detailed description of `dark_inter` can be found in HowTo perform dark subtraction with DARK_INTER.

11.3.4 `updown`

The purpose of this step is to correct for differences in up and down scans caused by memory effects (especially high flux levels bigger than 100 Jy, for bands two and four) and imperfect dark subtraction.
This correction is **not** implemented in the pipeline. Interactive tools to interpolate up/down scans are available (see HowTo do interactive up-down correction for extensive details on how to use this tool).

11.3.5 Responsivity Calibration

11.3.5.1 `respcal` `respcal` is the automated, pipeline responsivity calibration procedure. It is performed using the relative spectral response functions determined for each SWS AOT mode. Please refer to Section 7.3.4 in the ISO-SWS Handbook for details and the dc-document `respcal.pro` for syntax and usage.

11.3.5.2 `resp_inter` (`respcal`) `resp_inter` is written to replace `respcal`. It will reduce the amplitude of the fringe residuals through modification of the RSRF by matching the fringes in the RSRF with the observed fringes. It will shift and smooth or enhance the RSRF. It shifts the RSRF to compensate for wavelength shifts due to pointing offsets, source morphology or calibration drifts. The smoothing or enhancing is done to decrease or increase the amplitude of the RSRF fringes to compensate for the difference in resolution between the RSRF and the observational data.

Although `resp_inter` is an interactive tool, it has evolved over time such that its algorithms and default settings have become sufficiently stable to allow batch processing. In cases of doubt or to become available with the data and the nature of the corrections `resp_inter` can still be run in a highly interactive mode.

HowTo do RSRF correction interactively to minimise fringe residuals and HowTo minimise fringe residuals in SWS data describes in detail the associated problems and explains the use of `resp_inter` and of `fringes` which are often used together.

11.3.5.3 `fringes` (`respcal`) `fringes` models the residual fringes in the data after application of the RSRF by either `respcal` or `resp_inter`. The latter is preferred since in general the results after `fringes` are then better than when using `respcal`. This is a result of the fact that the fringe residuals are better defined after the corrections in `resp_inter`.

HowTo do RSRF correction interactively to minimise fringe residuals and HowTo minimise fringe residuals in SWS data describes in detail the associated problems and explains the use of `fringes` and `resp_inter`.

Note that `fringes` can also be used on a standard, cleaned or rebinned AAR (see section 11.5).

11.3.6 Flux Calibration

11.3.6.1 `fluxcon` `fluxcon` performs the conversion from signal in μV/sec to flux density in Jy. Detailed information on the flux calibration procedure is given in Section 7.3.4 of the SWS Handbook. For syntax and usage of `fluxcon` please refer to the dc-document `fluxcon.pro`.

11.3.6.2 plot_phot (fluxcon)

```
plot_phot, spd, sctbl, <phot>, <det> [, /all, /compare ]
```

Plots the photometric check data in the (dark current subtracted) spd and overplots the corresponding CAL41 data. The CAL 41 contains a reference photometric check which is used in the pipeline to determine the photometric gain (one gain for each band) of the observation. <phot> determines which photometric check is used. In the pipeline the default is the first. <det> is the detector number. When /compare is used a surface plot of the photometric check and the scaled CAL41 data of the whole band (of which <det> is part of) is made. When /all is used a multiplot for all detectors of the band is made. Useful in case a photometric check is suspect, see section 11.7.3

Data points flagged as bad by the system are marked. See the dc-document plot_phot for colour-code information and additional help. If these are the only bad points and the photometric checks look sensible then standard routines will handle the bad points appropriately. If the used photometric check has problems which are severe enough to affect the ultimate calibration, such as large spikes not already flagged, the routine PATCH_SPD might help.

11.3.7 Velocity Correction

To ensure a heliocentric wavelength calibration a velocity calibration must be applied. Please refer to the ISO-SWS Handbook for more details and the dc-document velcor.pro for information, syntax and usage.

The ISO velocity towards the target is taken from the SPD header. VELCOR displays the velocities TREFDOP1, TREFDOP2 and TREFDOP3 at the start, middle and end of an observations:

```
Message(M): Velocity components :       -21.510000      -21.490000      -21.460000 [km/sec
Message(M): Velocity correction of :       -21.490000 [km/sec]
```

The value at the 2nd reference time (TREFCOR2) is used for correction. Typical velocities are in the order of +30[km/sec]. The velocity correction is especially important in the case of Fabry-Perot observations.

11.3.8 Extracting AAR data

After all corrections to the SPD have been made, the AAR data can be extracted from the SPD.

```
aar = extract_aar( spd )
```

This will extract all valid data. This is data with a proper wavelength assignation and with SW, LW or FP runflag set. extract_aar will fill the AAR structure and copy all relevant information from the SPD. Additionally it will define a line and scnt in the AAR. For AOT-1 data the line number will follow the AOT-band order and the scnt will always be 1. For other AOT's they will be defined according to the order in which they occur in the SPD. This has the disadvantage that in some cases the same spectral line is listed in the AAR with different line numbers instead of the same line number with a different scnt. This is a result of the optimisation applied by the AOT-logic in the preparation of the observation. In OSIA it is possible to reassign the line and scnt into a more logical order.

```
aar = update_linecount( aar, filename=filename )
```

This will redefine the line and scan numbering and write the results to file. The line numbering starts at the lowest wavelength of the first aot-band and increases with wavelength and aot-band. Any line and scan from the same aot-band covering the same wavelength range is given the same line number with increased scan numbers.

Only valid points are extracted from the SPD. All the data points with zero wavelength.g. dark measurements) and all points flagged as NODATA (e.g. photometric checks after `fluxcon`) are skipped. SWS06 reference scans are also skipped unless the keyword `all` is specified. However, masked and glitched data points are forwarded to the AAR.

Note that an AAR can be extracted at any stage of the processing. This may sometimes be useful.

NOTE: To check that all the above steps have been preformed in a proper way, it is strongly recommended that the newly derived AAR is compared with the pipeline AAR (i.e. from the 'SWAAXXXXXXX.fits' file) or an aar derived with `daar`. All differences should be understandable as improvements to sophisticated processing or updated calibration tables.

11.3.9 Additional interactive processing routines

Two additional interactive routines are available within OSIA `mask_inter` and `patch_spd`. The order in which these tools may be used is given in Section 11.2.1. below we describe how these commands can be used. Please refer to the dc-documents `mask_inter` and `patch_spd` for more details,syntax and usage.

11.3.9.1 `mask_inter` The widget tool `mask_inter` is used to inspect the SPD and to mask out bad data points. It is usually called after dark subtraction and once again immediately before the AAR is extracted.

After dark subtraction its main tasks are:

• check dark subtraction: if the scans of the 12 detectors of a band are plotted simultaneously with a fixed, common y-scale, errors in the dark subtraction can be easily recognised. All scans should show a similar flux level, but the levels should not be identical as responsivity and flux calibration are still to be done.

• look for memory effects: if the up and down parts of a scan are overplotted with wavelength along the x-axis, memory effects become evident. In the case of strong flux changes at the beginning of or during a scan, up and down scans will differ significantly in a systematic way for all detectors of the band. The memory effects are strongest for bands 2 and 4.

• look for faulty detectors: during the mission it became apparent that individual detectors, especially of bands 2 and 3 can behave outside normal parameter (e.g. significantly increased dark current and hence significantly increased noise, a double structure or an untypically high glitch rate). Typical candidates for bad detectors are 15, 21, 31, 34 and 36.

• mask out bad data points: e.g. large spikes which have not been recognised as glitches, long glitch tails (which can corrupt the continuum level over a significant fraction of the scan length) and scans containing 'jumps'.

To display a scan, select a band and scan number and press 'disp_spd'. Clicking on the 'plotmode' button brings up a window to adjust plot scales. A fixed y-scale for all detectors is a good choice to check dark subtraction, to find extraordinarily nosy detectors or other strange features. After fixing the scale click on the 'w-mode' button and afterwards on the 'disp_spd' button to toggle between the SPD(time)-index and wavelength.

To mask out a specific section of a detector click on 'select' and on then on the plot of the detector of interest. This will replicate the plot of data from the detector in the lower left hand panel of the screen. Then click on 'mask' and use the left mouse button to select the region to be masked. If you see the same spike appearing in all 12 detectors of a band, you can use the 'do-to-all' button after masking out the glitch in one of the detectors.

A good check, whether features visible in all detectors of a band are real or not, is to look whether they appear at the same SPD-index (i.e. happen at the same time, likely to be spurious) or at the same wavelength (i.e. possibly to be a real feature). Additionally spurious features appear only in the up *or* the down scans, while real features should be present in the up *and* the down scans.

In addition at the pre-extract-aar stage mask_inter can be used to mask out strong glitches (including spikes and tails) and outliers. A very useful tool is the 'clip' button. When clicking this button data from *all* the detectors are plotted in the lower left hand window for *all* scans. The left mouse button can then define a series of boxes which encloses data points to be masked. Clicking with the right button returns to the main `mask_inter` tool.

Finally, saving the masked SPD should be considered f a significant amount of masking has been done. The masking performed by `mask_inter` is recognised by the post-AAR analysis routine. The effects of masking on the final result can be checked by using the 'ext_aar' and 'aarfilt' buttons which produce a filtered AAR of the specific section of the spectrum that has been selected with band, scan, order and x-range.

11.3.9.2 `patch_spd` `patch_spd` is a quick and rough tool to remove jumps, slopes and glitch tails from the data. See the dc-document `patch_spd` for more information, usage and syntax. `patch_spd` displays data of an individual scan (be it dark current, photometric check, data or reference scan) for a single detector. The data are plotted as white line. For updown scans, the mirrored data are overplotted in a white dotted line. The data from other detectors of a band may be overplotted on request. If the keyword `full` is used in calling the routine the whole SPD index range is displayed for the current detector.

To manipulate data a range must be selected. Selected data are then plotted in green. Patching should be used with caution. The first pass should be to mask bad data. Patching should only be used if a problem effects a significant part of a scan and low flux levels or S/N ratios require that as much of the data as possible must be used. Patching may also be used to mask regions where there is a signal jump or long glitch tail which destroys the baseline of scans at low flux levels producing errors in flat fielding and updown corrections.

11.4 Offband 4 data

The SPD contains additional data from the band 4 detectors. These are noted as AOT-band 4A, 4C and 4D. The wavelength ranges of these correspond roughly to those of the corresponding band 3 AOT-bands. The lower sensitivity and higher susceptibility to glitches of the band 4 detectors makes the quality of these bands usually less than those of band 3. They can however be useful to confirm a doubtful detection in band 3.

The offband 4 data can be accessed in the SPD just like any other band. To include offband 4 data in the AAR an `/offband` keyword has to be added to either the `DAAR` or `EXTRACT_AAR` call (see section 11.3.8).

```
aar = daar( spd, /offband )
aar = extract_aar( spd, /offband )
```

11.5 AAR processing

After having done the best possible reduction of the SPD and having extracted the AAR further processing may be required to optimise the results. This may involve further sigma clipping, flat-fielding, defringing, rebinning ..... and of course scientific analysis of the data which includes e.g. line and continuum fitting.

In HowTo Reduce a Grating AAR in the OSIA environment the options available within OSIA are described.

11.6 SPD manipulation routines

In table 12 a list of various OSIA routines is given which may be of use for the analysis of SPD data. No guarantee as to the actual usefulness and functioning of the routines is given as some may have been written back in the SWS stone ages. The user should therefore always first consult the OSIA dc-documents and the opinions of OSIA experts about these routines.

Or just try them and find out.

OSIA routine name	purpose
	SPD data access and assessment
CHECKPOORDETS	use dark noise to check for poor detectors
COUNT_GLITCHES	count the glitches per detector
MEANNOISE	compute mean and noise of data
PLOTSPD	plots the data of a SPD structure
PLOT_FULLFLAG	plot the flags of a SPD as a color image
PLOT_FULLSTATUS	plot the full SPD status as color lines
PLOT_PHOT	plots photometric check data
PLFLAG	plot the bits of the data flag plus meaning
SELECT	select parts of a SPD structure
	SPD data removal
CLEANSTRUCT	flag or remove unwanted data from the SPD
GLITCH_FILTER	flag glitched and/or saturated data
MASK_INTER	inspection of SPD and masking of data
SIGCLIP	sigma clipping on AAR or SPD
	SPD modification
LOW_SIGNAL	analysis of low-signal/deep-integration (AOT2) data
	(can also work on other types of data)
PATCH_SPD	patch SPDs, e.g. remove jumps, slopes, glitch tails
POINT_CORR	pointing and SSO tracking correction
	HowTo correct mispointed observations and Solar System Objects observations
SWS_FLATFIELD	flatfield corrections to AAR and SPD data
UPDOWN_INTER	highly interactive up-down correction

Table 12: SPD manipulation routines

11.7 Pipeline warnings and error messages

Most pipeline routines issue informative messages or warnings to the user. In the following section some of these are listed by pipeline processing step.

The level of the error messages can be one of the following:

Debug(D) Debug level, included for development and problem analysis.

Message(M) An informative message for the user.

Warning(W) A problem has been detected and reported to the user. The problem is caught inside the programme and the processing continues as normal.

Serious(S) A problem is discovered after which no standard processing is possible. The programme controls the manner in which the processing continues. It either returns to the caller level or continues with a non-optimal reduction.

Fatal(F) An unrecoverable problem is discovered, the processing stops and returns to the main level.

In OSIA the user can control the level of the warnings printed in the IDL window using the routine `set_error`. Only messages with an error level equal or higher than set by the user is printed.

```
set_error,mes_level='W'
```

In which case only messages with an error level W,S or F are printed.

11.7.1 DAAR shell

```
Serious(S): SPD input is not defined
Fatal (F): Could not open <FILENAME>
```

When one of these errors is issued the input SPD structure or filename is not proper.

```
Message(M): Your SPD was processed with OLP: OLP_10
```

Refers to the OLP version listed in the header. Note that when the SPD is rederived in OSIA the OLP version of the ERD is kept.

```
Warning(W): No AAR produced
```

The SPD contained no data with a valid wavelength.

11.7.2 DARK (including DYNADARK)

```
Warning(W): Dark current not reliable <det> - 13, 2191, 2223, 2467, 2497
```

Indicating the detector number and the index numbers of the first sample of the dark current blocks for which there are not enough valid dark current points.

```
Warning(W): Darkcurrent out of range for detectors <dets>
```

In dark the measured dark currents are compared to standard dark current values contained in the dark current calibration files CAL21_1, CAL21_2, CAL21_4 and CAL21_8. These contain dark current values derived from calibration measurements under controlled circumstances for each reset time. In many cases dark current measurements in an observation will differ from these due to memory effects or changes in the absolute zero level of the detector. In such a case this message is issued. It does not mean that there is necessarily a problem with the observation and if upon visual or another type of inspection no anomalies are seen this warning can be ignored.

The following warning may be issued by dynadark. They do not often occur and are mostly an indication that the limits of the memory model employed by dynadark are met than that there is a real problem. In most of these cases the result after dynadark has still improved. See the paper by Do Kester [4] for more details on the employed method and the meaning of the individual error messages.

11.7.3 FLUXCON

The most critical point in the flux calibration processing is the derivation of the photometric gain for the observation. The photometric gain is defined as the ratio of the internal photometric check with the reference photometric check in CAL41.

```
Serious(S): NO GRATING CALIBRATOR DATA
```

A photometric gain of 1 is applied.

```
Warning(W): PHOTOMETRIC GAIN IS OUT OF LIMITS, BAND <band>
```

The photometric gain is less than 0.5 or more than 1.5. Gain offsets of up to 25% are not uncommon. Very large gains offsets however could mean there is a problem and the user should manually check the photometric check in his data. The large or small gain is applied unless the following error is issued.

```
Serious(S): Zero or negative gain for det:  <det>
```

The derived photometric gain is less than 0.1. Something is really wrong and the derived gain is not used but instead a gain of 1 is assumed.

In case of doubt use the `plot_phot` routine, see section 11.3.6.2.

11.7.4 RESPCAL

```
Warning(W): No valid cal19 data, data are treated as AOT6
```

For AOT-1 data. Usually means that the AOT speed could not be determined from the SPD header. In that case the user should get the correct AOT-speed and enter that via the command line either to `RESPCAL` or `DAAR`. If this is omitted the data are corrected with the nominal RSRF.

```
Fatal(F): No valid AOT band left
```

Issued when in an AOT-1 no grating AOT-bands are defined. This can happen when an observation is affected by a telemetry drop or when a large part of the data has been selected out or flagged as bad data within OSIA in the processing up to `respcal`.

11.7.5 VELCOR

```
Warning(W): No valid velocity data in SPD header
```

No velocity correction is done. Make sure to update the SPD header.

```
Message(M): Velocity components :  -29.500000 -29.460000 -29.420000 [km/sec]
Message(M): Velocity correction of :  -29.460000 [km/sec]
```

The ISO velocity components read from the SPD header and the single average velocity used to correct the SPD data.

11.7.6 EXTRACT_AAR

```
Message(M): AAS_EXTRACT_AAR gave <AARSIZE> points
```

With `AARSIZE` the number of points in the extracted AAR

12 HowTo perform dark subtraction with DARK_INTER

This document[2] describes how to perform sophisticated dark subtraction with the OSIA routine DARK_INTER.

Dark subtraction is a very important step in SWS data reduction, particularly so whenever dark currents are significant in comparison to the flux level of individual scans and bands. Accurate dark current subtraction becomes essential in measurements where the flux level is generally low.

DARK_INTER allows one to perform highly interactive dark subtraction with careful determination of the dark currents and a visual success control. As the quality of the result is driven by visual inspection and, if necessary, detailed dark correction for individual scans, using this tool can be rather time-consuming. However, especially in case of low flux levels, using DARK_INTER is strongly recommended and can significantly improve the final result. Using DARK_INTER can also be of advantage in the case of medium and high flux levels, as the automatic dark subtraction (pipeline module DARK) may not be optimal in all cases. Such sub-optimal subtraction may occur when, for example, cosmic rays hit the detector during or close to a dark measurement producing spikes with long tails or if dark measurements are heavily affected by memory effects.

Careful analysis of the data presented in DARK_INTER additionally helps the user to become familiar with the dataset and to recognize problems such as single/multi detector jumps, strong cosmics with long after-effects and the influence of memory effects (already seen at an earlier stage of data reduction).

12.1 Introduction

The OSIA module DARK_INTER facilitates highly accurate dark subtraction for all SWS AOTs by virtue of its interactivity.

It is the interactive counterpart of the pipeline routine DARK (*see the dc-document* dark), which is normally executed after the pipeline module ANTIMEM, as second step on the way from SPD to AAR HowTo process SWS data from SPD to AAR. This document gives detailed instructions on how to use DARK_INTER. At first glance, DARK_INTER seems to be a very complicated tool but after some experience the user will find using the module is rather simple. Newcomers to DARK_INTER should read these instructions carefully.

A key issue is to understand how DARK_INTER works in general. Dark subtraction with DARK_INTER is done in two steps: first the dark measurements are refined, then the real dark subtraction is performed.

This means for example, for a single scan, that in the first step a kappa-sigma clipping is applied to the dark measurements before and after the scan to remove outliers. Additionally, glitched dark points are removed, which are known to be bad data. In a second step, the actual dark current is evaluated by a linearly interpolating between the before and after dark measurements. The dark current is then subtracted from the flux values measured during the scan.

Various methods are available in DARK_INTER to refine dark measurements and to perform dark subtraction for scans and photometric checks. These methods can not only be applied to a single dark, scan or photometric check as measured by a single detector, but also simultaneously to all 12 (FP: 2)[3] detectors of a band, and even to *all* darks, scans or photometric checks of *all* detectors of a band.

Dark current subtraction is *always* performed on input fluxes. This enables the trial and testing of alternative dark subtraction methods. It also allows special refinements to be made for problematic cases after a standard method has been applied to all scans or photometric checks of a band. The *final* refinements and method applied to the measurement defines the output spd.

[2] This document is based on notes and recipes by Dietmar Kunze, Eckhard Sturm and Michele Thornley.
[3] Only one detector of each Fabry-Perot pair, namely 49 and 51, record science data. The other ones are spare.

12.2 How to use DARK_INTER

DARK_INTER is called as an IDL function. It has two arguments, *spd* and *sctbl*, which are the names of the input SPD and its associated scan table structure.[4] To call DARK_INTER, enter:

spd_D = **dark_inter**(*spd, sctbl*)

The scan table *sctbl* should be derived from the SPD[5] and printed in advance with the OSIA commands MAKE_SCTBL and PRINT_SCTBL HowTo process SWS data from SPD to AAR. If the names of the input and output SPD are different, an output SPD with the specified name (e.g. *spd_D* in the example above) is created, otherwise the original input SPD is overwritten.

NOTE: server or window systems usually are configured to retain plot windows after scrolling or starting a screen saver. If the draw window is not retained (as on Multia terminals or often with Linux PCs), the **/multia** keyword has to be used as additional argument in the call of DARK_INTER, for IDL to handle the plot storage directly.

Before using DARK_INTER one should have a careful look at the scan table. There, the sequence of darks, scans and photometric checks for the short wavelength (SW1/2: Band 1+2) and long wavelength (LW1/2: Band 3+4 or FP1/2: Band 5+6) sections are listed together with additional information HowTo get a quicklook at your observation. The reset time ('tr') and gain settings in the scan table must be the same for the dark measurements and the scan or photometric check from which they are subtracted. DARK_INTER automatically makes available only those darks with matching reset time and gain settings. Also, check the scan table to determine which band/scan combinations hold actual data: scans which show '0.000' for both wave_min and wave_max do not contain meaningful, calibrated science data. Such scans do not need to be dark subtracted.

12.2.1 The DARK_INTER main window

After calling DARK_INTER the main window pops up (*see fig. 5*). It consists of three panels: two multiple plot windows on the left and right side and the control board in the middle.

The control board contains the detector selection at the top and the scan/phot selection in the middle. Additionally there are two scrollable text windows and a column of buttons in the lower part of the panel. By default, detector number 1 is selected. The current selection is displayed in the detector number entry field. The selection can be changed by moving the detector slider, clicking the [previous] or [next] (detector) button, or by entering a new detector number (NOTE: any entry of a valid detector number [1,..,52] must be confirmed by pressing the [RETURN] key). The number of the current detector determines, which of the six AOT bands is selected (band 1: 1-12, band 2: 13-24, band 3: 25-36, band 4: 37-48, band 5: 49+50, band 6: 50+51). The data of the selected detector are plotted in the left panel. Whenever a new detector is selected, the plots are updated.

The left panel is a scrollable plot window, which itself is divided into three parts. The upper plot shows the flux of the selected detector plotted as a solid line against SPD index[6]. Color coding is used to indicate different parts of the data: dark current measurements are plotted in yellow, scientific scans in green and photometric checks in cyan. Scientifically non-usable scans, which have wavelength zero assigned throughout the scan, are plotted in deep blue. Any other data, such as data in between darks, scans and photometric checks or data at the beginning and the end of an observation, are plotted in white (see Table 13). The x- and y-range of the upper plot are chosen by default such that all data, which have been recorded by the selected detector during the whole observation, are displayed. Use the slider at the bottom of the left panel to see the hidden parts of the window.

[4] The names for the *variables* are only representative and can, of course, be chosen by the user.

[5] if the sctbl argument is omitted, DARK_INTER re-derives the scan table from the spd, however we strongly recommend that the scan table is derived and printed for reference for the entire reduction process

[6] All 52 SWS detectors record data during the whole time of any SWS observation.

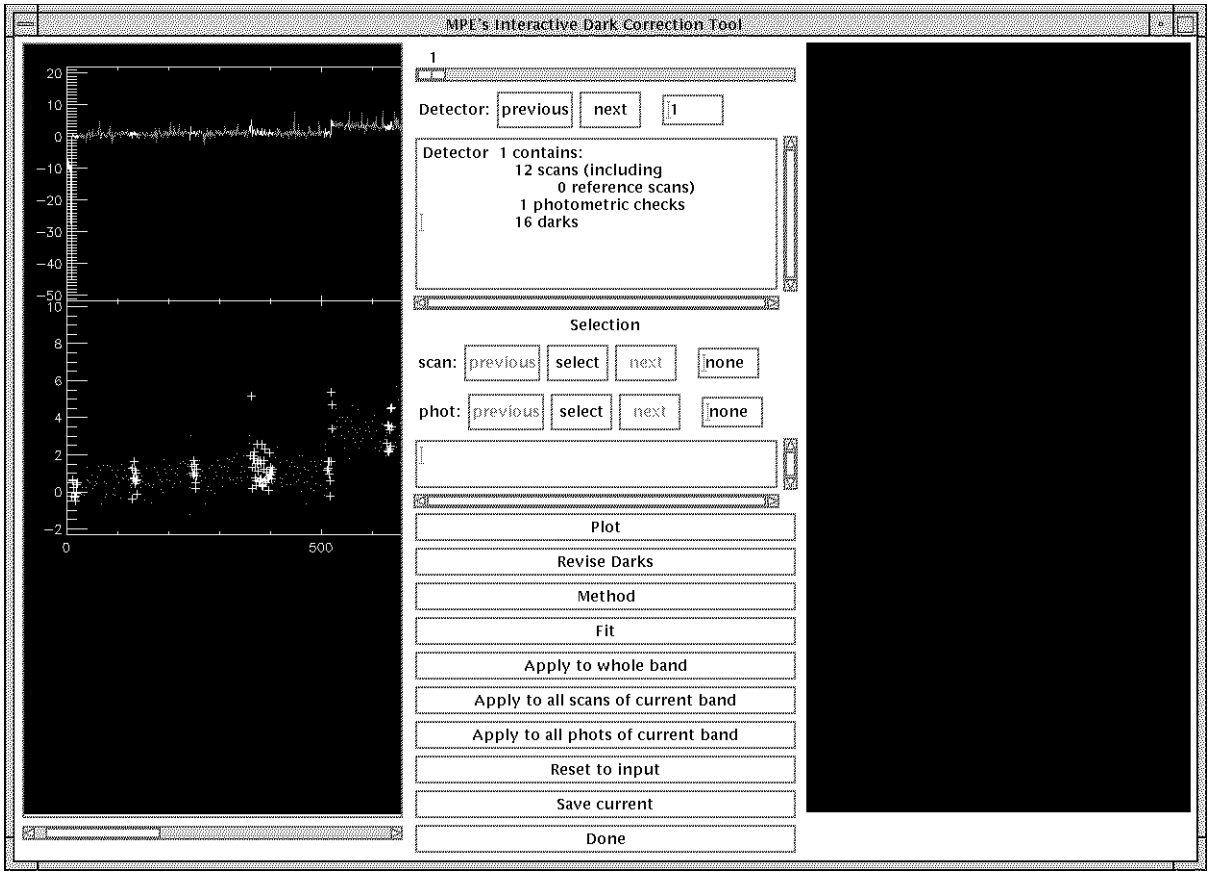

Figure 5: The DARK_INTER main window for an SWS02 observation after setup.

The middle plot of the left panel shows again the flux of the selected detector plotted against SPD index, but now zoomed-in to dark level and using different plot symbols. The y-range of this plot is defined such that all dark values, measured by the current detector during the whole observation, are displayed. The individual dark points are plotted as crosses, any other data points as dots. The x-scale and the color coding is the same as in the upper plot. The third, lowest plot on the left panel and the whole right panel are filled at a later stage of using DARK_INTER.

The text window between detector and scan/phot selection shows corresponding information from the scan table, like the number of scans, photometric checks and darks for the current detector/band (and after a scan or phot has been selected also information on individual scans/phots). The second text window is reserved for error messages. More details on the scan/phot selection and the various buttons in the lower part of the control board are given in the following sections.

To familiarise users to the dark inter processing steps a general recipe, how to use DARK_INTER, follows:

- A: select the first detector of a band (1, 13, 25, 37, 49 or 51)
- B: use the 'plot' window to adjust plot scales (if necessary)
- C: 'revise dark' measurements of all detectors of the current band:
  - C1: select method: 'clipping (mean)' with 'clip repeat', 'kappa: 3' and 'glitches' (ignored)
  - C2: [apply it to all darks of current band]
- D: perform dark subtraction for all scans and photometric checks of current band:

Color Code	Meaning
green	all/selected science scans
deep blue	scans with zero wavelength
cyan	all/selected photometric checks
yellow	all/selectable dark currents
white	any other/unselected data
red	selected dark measurement
light blue	masked dark points
white box	glitched dark points

Table 13: The DARK_INTER color code.

  - D1: apply standard method scan by scan (or phot by phot) to the whole band:
    * a: select the first/next non-zero scan (or phot)
    * b: use the 'method' window to select 'prev+next/linear' for scans (or 'prev/mean' for phots)
    * c: press [fit] to apply the standard method to the selected scan (or phot) of the current detector
    * d: [apply (the method) to the whole band], i.e. to all 12 detectors of the current band
    * e: if necessary, re-revise the used dark measurements and/or select a non-standard method for dark subtraction and apply the new darks/method by pressing [fit] to the selected(!) detector
    * f: note down for later elimination, if weird effects, like detector signal jumps, bimodal data point clouds or extraordinary high dark (and hence noise) levels appear
    * g: repeat this procedure for all scans *and* all photometric checks of the current band
  - D2: check standard method detector by detector and make improvements, if necessary:
    * a: select the next detector of the current band
    * b: check applicability of standard method by visual inspection for all scans and photometric checks of the current detector
    * c: if necessary, select a scan/phot, select the appropriate standard method and press [fit] to verify the applied method for the current scan/phot
    * d: to improve dark subtraction, re-revise the used dark measurements and/or select a non-standard method for dark subtraction and press [fit] to perform dark subtraction with the new darks/method for the currently selected(!) scan/phot (do *not* use [apply to the whole band])
    * e: note down for later elimination, if weird effects, like detector signal jumps, bimodal data point clouds or extraordinary high dark (and hence noise) levels appear
    * f: repeat this procedure for all detectors of the current band
  - D3: save current status
- E: repeat the whole procedure for all other bands

Detailed information on the tasks addressed in this recipe is given in the sections below.

12.2.2 The 'plot' window

The 'plot' window is a separate window (*see fig. 6*), which is used to adjust the plot range of the plots in the left panel of the DARK_INTER main window (*step B of the recipe*). It is opened by clicking on the [plot] button of the DARK_INTER control board. The text field at the top of the 'plot' window contains the maximum 'data range' in x (= SPD index) and y (= flux density in [μV/s]) for the current detector. The 'xmin', 'xmax' and 'ymin', 'ymax' fields below contain the plot ranges of the middle left plot of the DARK_INTER window as given at start-up. Most of them update on clicking the plots in the left part of the DARK_INTER window. In addition, there is a row of buttons at the bottom of the 'plot' window whose functionalities are subsequently described.

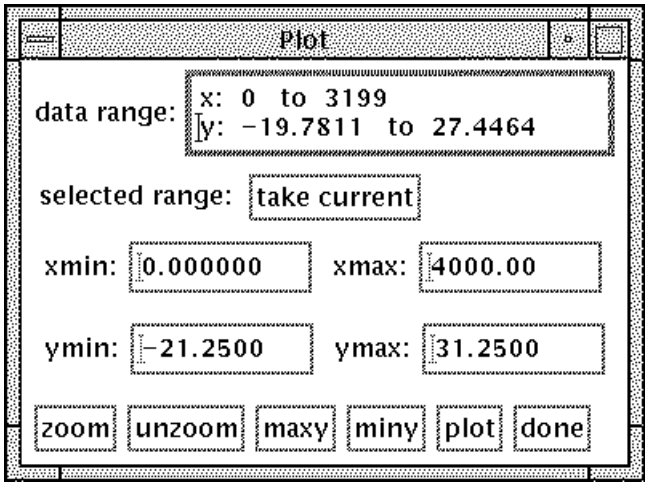

Figure 6: The DARK_INTER 'plot' window.

Some of the buttons apply fixed default ranges to the plots, which are defined when a detector is selected. The [maxy] and [miny] buttons modify the y-range of the plots. Clicking on [maxy] applies the original (=maximum) y-range of the upper left plot to all plots in the left DARK_INTER panel. By clicking on [miny], the original zoomed-in y-scale of the middle left plot is used. These buttons allow one to toggle between an unzoomed and zoomed-in view of the data. The [unzoom] button itself resets the plot ranges to their original values, as they were when the current detector was selected.

The x- *and* y-range can be modified with the [plot] button of the 'plot' window. This button applies the ranges displayed in the 'xmin', 'xmax' and 'ymin', 'ymax' fields to all plots in the left DARK_INTER panel. The ranges can be modified by direct user input, with the [take current] or [zoom] buttons. The [take current] button takes the current ranges of the middle left plot of the DARK_INTER main window. The [zoom] button creates a cross-hair in the upper left plot. Click twice in this plot, to define two opposing corners of an x-y-range box, then press the [plot] button in the 'plot' window to actually update the plots with the new ranges.

The 'xmin', 'xmax' and 'ymin', 'ymax' values are updated automatically with the new ranges of the middle left plot when a *new* detector of a *new* band is selected. However, they are not changed if only a new detector is selected but the band remains the same. Therefore, the 'plot' window can easily be used to apply fixed plot scales successively to all detectors of the current band.

The [done] button of the 'plot' window closes the 'plot' window temporarily. However, it may be useful to keep the 'plot' window open during the entire DARK_INTER session to re-adjust plot ranges, if necessary. Click on the [plot] button of the DARK_INTER control board to re-open the 'plot' window or to bring it to the foreground, if it is hidden behind the main window.

12.2.3 The 'revise darks' window

An important step of dark subtraction with DARK_INTER is the refinement of the dark measurements. This is done with the 'revise darks' window (*see fig. 7*), which is opened by clicking on the [revise darks] button of the DARK_INTER control board. Dark refinement appears at several places in the recipe given on page 59. The following section describes the refinement of *all* darks of *all* detectors of a band with a standard procedure (*step C in the recipe*). The next but one section describes how to refine individual dark measurements, which might be necessary at a later stage of DARK_INTER (*steps D1e and D2d*).

12.2.3.1 Revise all darks of current band Dark subtraction with DARK_INTER is done band by band. It starts, for each band, usually with a standard refinement of all dark measurements of the current band (*step C in the recipe*). The refinement method is defined in the middle part of the 'revise darks' window. The recommended setting is 'first x' with 'x: 1', 'clipping (mean)' with 'clip repeat', 'kappa: 3' and 'glitches' (ignored). Press the corresponding buttons to activate these 'cut methods'. Due to the default setting, 'first x', the first point ('x: 1') of each dark measurement in the current band is masked.[7] Additionally, all dark points which have been recognized and flagged as 'glitches' are masked and are ignored during the determination of the dark current.

The main step of the standard refinement procedure is a kappa-sigma 'clipping' around the mean value of each dark with a 'kappa of 3' (or the user defined value of kappa). This clipping masks all dark points outside the clipping limits of kappa times the standard deviation around the mean value. Depending on the number of original and remaining points in a dark measurement, the clipping is repeated several times ('clip repeat'), to mask outliers successively. The whole procedure is [applied to all darks of current band] by pressing the corresponding button at the bottom of the 'revise darks' window. This initial standard refinement of all darks of a band usually does a rather good job, so that special treatment of individual darks later on is only necessary for a few of them.

12.2.3.2 Revise individual dark measurements To revise an individual dark measurement (*like proposed in step D1e and D2d*) the dark has to be selected. This is done with the selection buttons or the dark number entry field at the top of the 'revise darks' window. If no dark is selected ('dark: none'), the [previous] and [next] button select the first dark of the current detector. Otherwise, clicking these buttons selects the previous or next dark to the one shown in the dark number entry field box. Any dark measurement of the current detector can be selected by clicking on the [select] button and subsequently clicking on the desired dark in the middle left plot of the DARK_INTER main window. Additionally, any dark can be selected by entering its specific number in the entry field (NOTE: any entry of a dark number must be confirmed by pressing the [RETURN] key).

When a dark is selected its color in the left panel of the DARK_INTER main window turns from yellow to red. In the right panel, the selected dark measurements from each detector of the current band (grating: 12, FP: 2) are displayed. The dark of the currently selected detector is plotted enlarged with respect to the other detectors in the band. Glitched dark points are surrounded by a yellow square. Masked dark points, which are later ignored during the determination of the dark correction, are plotted in gold/dark yellow.

The 'cut method' section of the 'revise darks' window offers various methods to refine dark measurements. The 'first x' and 'last x' options mask, if selected, the first x and last x data points of the current dark measurement. 'first x' with 'x: 1' is default. The 'glitches' option, in addition, masks all dark points flagged as glitches. The effect of these options on the current detector is immediately displayed in the right panel, as soon as the option is selected. To remove such masking, toggle with the corresponding button. The currently displayed masking becomes permanent, if the [apply] button of the 'revise darks' window is pressed.

The 'first x' option is especially useful if a dark measurement, due to memory effects, suffers from a strong illumination in the preceding scan or photometric check. The best guess for such a dark is then to use only the last few points of the exponentially decaying signal of the dark. For example use 'first x' with 'x: $n_{dark} - 3$', with n_{dark} being the total number of points of the dark. To apply such a value, enter the new value in the 'x:' entry field, then toggle with the 'first x' button (i.e. switch the button off and on).

Dark points of the currently selected detector can also be masked or unmasked with the [mask] and [unmask] buttons. After clicking one of these buttons a cross-hair appears in the plot for the current detector (enlarged plot!) in the right DARK_INTER panel. Click twice in this plot window to define a selection box. All dark points in the box will be masked or unmasked, respectively.

Another option is 'clipping (mean)', which initiates a kappa-sigma clipping around the mean of the dark cloud with the specified kappa value and masks the outliers. Clipping is activated if the 'clipping (mean)' option is selected (click the preceding button). To display the points to be clipped for the current detector, press the [clip] button. The two vertical lines will then appear in the enlarged plot in the right panel for the current detector showing the specified clipping limits.

[7] The pipeline as only step ignores the first 3 data points of any dark.

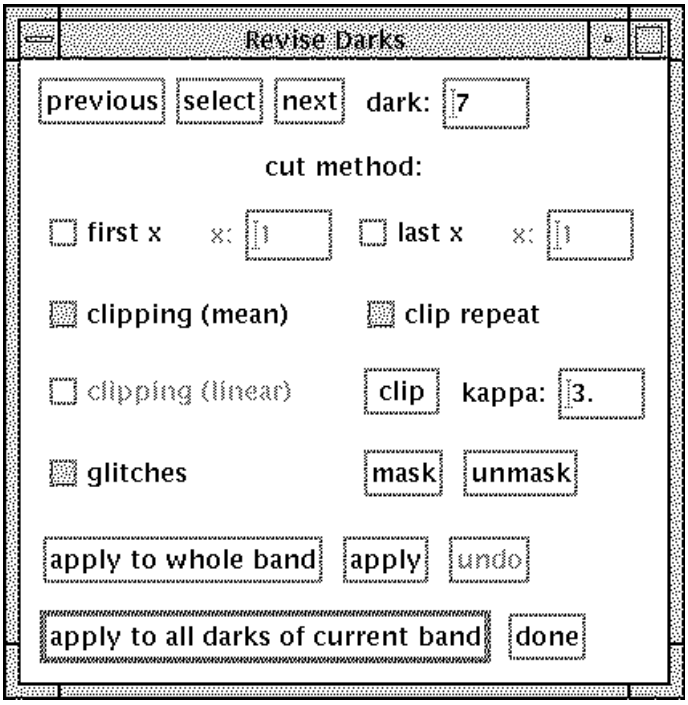

Figure 7: The DARK_INTER 'revise darks' window.

Thus, all points lying above or below these lines are masked. The [undo] button can be used to reset the masking. If the [clip] button is pressed several times without clicking the [undo] button in between, the clipping is repeated. Finally use the [apply] button to make the masking permanent.

A typical standard method for dark refinement of individual darks is 'clipping (mean)' with 'kappa: 3' combined with 'first x' with 'x: 1' and 'glitches' (ignored). **NOTE: Before selecting another dark or leaving the 'revise darks' window, the [apply] button has to be pressed, to make the masking permanent!** If modifications have been made to a dark measurement, which neither have been confirmed by pressing [apply] nor [undo]ne, a 'confirm' window pops up and asks the user to do so. This window appears as soon as a new dark is selected or the current dark is used for dark subtraction.

With exception of the [mask] and [unmask] options, which are detector specific, the refinement methods discussed above can not only be applied to the currently selected detector, but also to all detectors of the current band. This is done by the [apply to whole band] button. As all detectors of a band are made out of the same material, perform the same measurement at the same time and have the same illumination history, applying the same method to the whole band is appropriate. The [undo] button can be used to reset such a masking, as long as the [apply] button has not yet been pressed.[8] If clipping is selected, the clipping for all detectors of the band can be repeated by pressing the [apply to whole band] button several times, without clicking the [undo] button in between. Remember, that the final masking must be made permanent, by pressing the [apply] button.

As discussed in the preceeding section, dark revision has to be done for all darks of all detectors of a band before they are used for dark subtraction. The [apply to whole band] button can be used instead of the [apply to all darks of current band] super-button to apply a standard refinement method to all darks of the current band. After the method has been defined, the first dark of the band is selected and the method (clipping!) is repeatedly [applied to the whole band]. If the masking is satisfactory, the [apply] button is pressed to make the refinement permanent. Then the [next] dark is selected. This

[8] If too many dark points of a dark measurement have been masked for various detectors of the current band and the masking has already been applied, the 'first x' button can be used to *un*mask all dark point of all detectors of the current band. The procedure is: enter 'x: n_{dark}', switch 'first x' off and on, then press [undo], switch 'first x' off and finally press [apply].

procedure is repeated for all darks of the band. The advantage of this procedure compared to the super-button is, that the masking for each dark can be checked and corrected, if necessary. This might be useful in case of unusual data.

The 'revise darks' window is closed with its [done] button. The window may stay open during the entire DARK_INTER session. Click on the [revise darks] button of the DARK_INTER control board, to re-open the 'revise darks' window or to bring it to the foreground, if it is hidden behind the main window.

12.2.4 The 'method' window

Before the actual dark subtraction is performed a subtraction method has to be defined (*steps D1b, D1e and D2d*). This is done in the 'method window' (*see fig. 8*). To open this window, click on the [method] button of the DARK_INTER control board. As the absolute signal level of individual detectors before dark subtraction and the signal noise depend on reset time and gain setting, only darks which have the same reset time and gain setting as the scan or photometric check under consideration can be used for dark subtraction. The 'method' window defines which darks are to be used. In addition, it determines the function to describe the dark current between the selected darks.

The upper part of the 'method' window offers four standard methods to determine the dark current for subtraction: the mean of the dark measurement preceding the scan/photometric check ('prev/mean'); the mean of the dark measurement following the scan/photometric check ('next/mean'); the mean of the dark measurements *both* before *and* after the scan/photometric check ('prev+next/mean'); calculate a linear interpolation between the dark measurements preceding and following the selected scan/photometric check ('prev+next/linear').

Grating scans usually have 'proper' preceding and following darks. Therefore, the standard method to evaluate the dark current is to perform a linear interpolation between the before and after darks: 'prev+next/linear'. Photometric checks are much shorter than scans and usually have only 'proper' preceding darks. The standard method for such measurements is to use the constant mean value of the preceding dark: 'prev/mean'.

'Prev' and 'next' denote the darks immediately before and after a scan or photometric check. Beside these, all other darks with proper reset time and gain setting can be used for dark subtraction. To select any other dark measurement to be used in the evaluation of the dark current click on the [cursor] button in the 'method' window and then on the appropriate dark measurement in the middle left panel of the DARK_INTER window. Once selected its color turns from yellow to red. By clicking on [cursor] and then on an already selected dark in the middle plot window de-selects this dark (the color of the points from this dark then returns to yellow). The [prev]ious and [next] dark can also be de-/selected by toggling with the corresponding buttons. All selected darks are taken into account in the dark subtraction. Depending on the method they either contribute to the 'mean' or are used for 'linear' interpolation.

If the available darks do not provide a good dark measurement to be subtracted, a spline fit may be used to define an arbitrary dark current. To do so, choose 'spline' in the 'method' window and follow the procedure outlined for spline fitting in the following section. NOTE: The spline fit must be done for each detector individually. Therefore, use it only if necessary.

Click [done] to close the 'method' window. This window may stay open during the entire DARK_INTER session.

12.2.5 Perform dark subtraction

The actual dark subtraction is done by clicking the [fit], the [apply to whole band] or the [apply to all scans(/phots) of current band] button of the DARK_INTER main window (*see fig. 9*). These buttons treat a different number of scans or photometric checks at the same time.

The [fit] button applies a previously defined dark subtraction method to the currently selected scan or photometric check of the currently selected detector (*steps D1c, D1e, D2c and D2d*). To select a science scan or photometric check click on the corresponding scan/phot [select]ion button and subsequently on the desired scan or phot in the middle left panel of the DARK_INTER main window. Another possibility is to enter the related scan or phot number in the corresponding

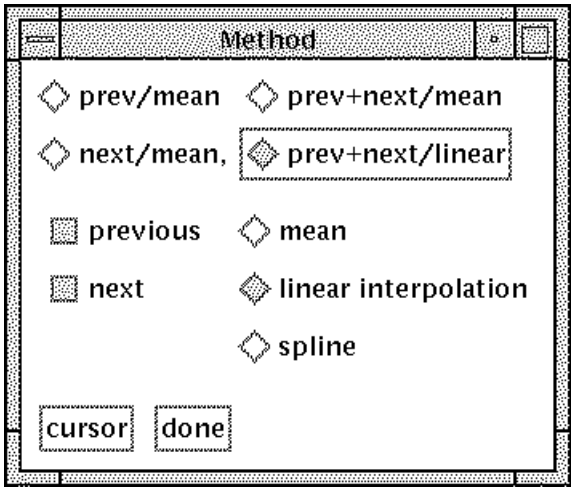

Figure 8: The DARK_INTER 'method' window.

entry field (NOTE: any entry of a scan or phot number must be confirmed by pressing the [RETURN] key). If a scan or phot has already been selected, skip to the preceeding or following one by pressing the corresponding [previous] or [next] button.

As soon as a scan or photometric check is selected, all other data in the plots in the left DARK_INTER panel are displayed in white. Only the selected scan or photometric check is highlighted in green or cyan respectively, and centered in the left panel.[9] Additionally all dark measurements, which have the same reset time and gain setting as the considered scan or photometric check, are colored yellow. These are the dark measurements which principally can be selected for dark subtraction. As the dark current might change with time, it is recommended to use only the darks close the considered measurement for dark subtraction.

With the 'method' window, the dark subtraction method can be adjusted to the available dark measurements. If a specific method is selected, like 'prev+next/linear' for scans or 'prev/next' for photometric checks, the darks to be used are colored in red. Using the [cursor] button of the 'method' window additional (yellow=selectable) darks can be selected. Click first on the [cursor] button and then on the desired dark in the middle left panel. With the same method already selected darks, including [prev] and [next], can be de-selected.

If the [fit] button is pressed, dark subtraction is performed with the selected darks and method for the current scan or photometric check and detector. The calculated dark current is displayed as a white line in the middle left panel[10]. The result of the dark subtraction is displayed in the third, lower left panel. There the dark fit of the middle left panel corresponds to the zero-y baseline. The dark fit and dark subtraction are displayed for the whole x-range, but only applied to the data of the selected scan or photometric check (colored green or cyan respectively)! Different methods and dark measurements can be tried. The final method used determines the output of DARK_INTER. Previous fits are left on the plot as a guide. The plots can be refreshed using the 'plot' window.

In case of low fluxes, sometimes detector signal jumps are present in the data. If these are simply horizontal steps, then it is advisable to use either the mean of the previous or next dark to fit the longest subsection and then mask out the other subsection later in the MASK_INTER tool or correct the offset using PATCH_SPD HowTo process SWS data from SPD to AAR.

In the 'method' window a 'spline' option can also be selected. This option can only be used in combination with the [fit]

[9] Make sure that the entire scan or photometric check can be seen in left windows. If necessary, use the 'plot' window to adjust the plot parameters, or use the slider bar at the bottom of the left panel to adjust the horizontal range.

[10] If you get strange fits or error messages when trying to subtract a dark, it may be because the dark was not selected properly. Go back to the 'method' menu and toggle your dark choices until they appear in red in the left plot windows.

button. To use this fitting algorithm select spline in the method window and press [fit] in the main DARK_INTER window. Instructions on how to proceed are given in the DARK_INTER dialog box in the center of the window. This method is controlled using the mouse buttons allowing definition of dark points for spline fitting, fitting the spline function and iterating until an optimal fit is obtained. Spline points should be defined on the middle left plot using the left mouse button (NOTE: closely spaced spline points are likely to cause a highly structured fit, while widely spaced points are often better). If a spline fit is necessary for all 12 detectors of a band, the fit has to be repeated for each individual detector: increment the detector number, select the scan and perform a new spline fit.

All other available fit methods for the current scan or photometric check can be applied to all detectors of the current band with the [apply to whole band] button (*step D1d*). The result of the dark subtraction for the 12 (FP: 2) detectors of the current band is then shown in the right panel (*see fig. 9*). [fit] and [apply to whole band] are used in the recipe given on page 59. The main outline is given here again:

- Select first detector of a band. This selection defines the current band.
- Revise all darks of all detectors of the current band with the standard method.
- Subtract dark currents for all scans and phots with standard fit method:
  - Select the first non-zero scan.
  - Select standard fit method 'prev+next/linear' (or 'prev/mean').
  - Press [fit] and [apply to whole band] to apply the standard method for the current scan to all detectors of the current band.
  - Check dark refinement and applicability of the standard method for the selected (=first!) detector of the band. If necessary, re-revise the used darks individually or select a non-standard fit method and apply it with the [fit] button. Do *not* [apply (this special treatment) to the whole band]!
- Repeat these steps for all scans and photometric checks of the band.
- Check dark refinement and standard fit method for the other detectors:
  - Increment detector number.
  - Check dark refinement and applicability of the standard method for all scans and photometric checks of the currently selected detector. If necessary, select a scan or photometric check, re-revise the darks or select a non-standard fit method and apply it with the [fit] button. Do *not* [apply (this special selection) to the whole band]!
- Repeat these steps for all detector of the current band.
- Select the first detector of the next band and repeat the whole process, for all four bands.

NOTE: A key issue is that each individual scan or photometric check of each detector and band is visually inspected after a default method has been, more or less, blindly applied (*step D1e and D2a-f!*). If necessary, the dark measurements must be re-refined with the 'revise darks' window or a non-standard subtraction method must be selected with the 'method' window. Subsequently, the [fit] button has to be pressed in order for the dark subtraction to be performed again for the selected scan or photometric check of the selected detector. The actual dark subtracted data are only displayed in the lower left plot of the DARK_INTER main window. The middle and the upper left plot always show the input data! The output SPD contains for all treated scans and photometric checks the dark subtracted fluxes, as they result from the last dark correction applied to these measurements.

The [apply to all scans of current band] and [apply to all phots of current band] buttons apply a previously defined fit method subsequently to all scans or photometric checks of all detectors of the current band. This is an even more 'blind' application of a fit method. A subsequent careful visual inspection is mandatory and, if necessary, a detailed refinement of the dark subtraction of *every* scan and photometric check of *every* detector of the current band is strongly recommended. These buttons can be used to automize step D1 ('apply standard method') of the recipe. Step D2 ('check dark subtraction') has then to start with the first detector of the band.

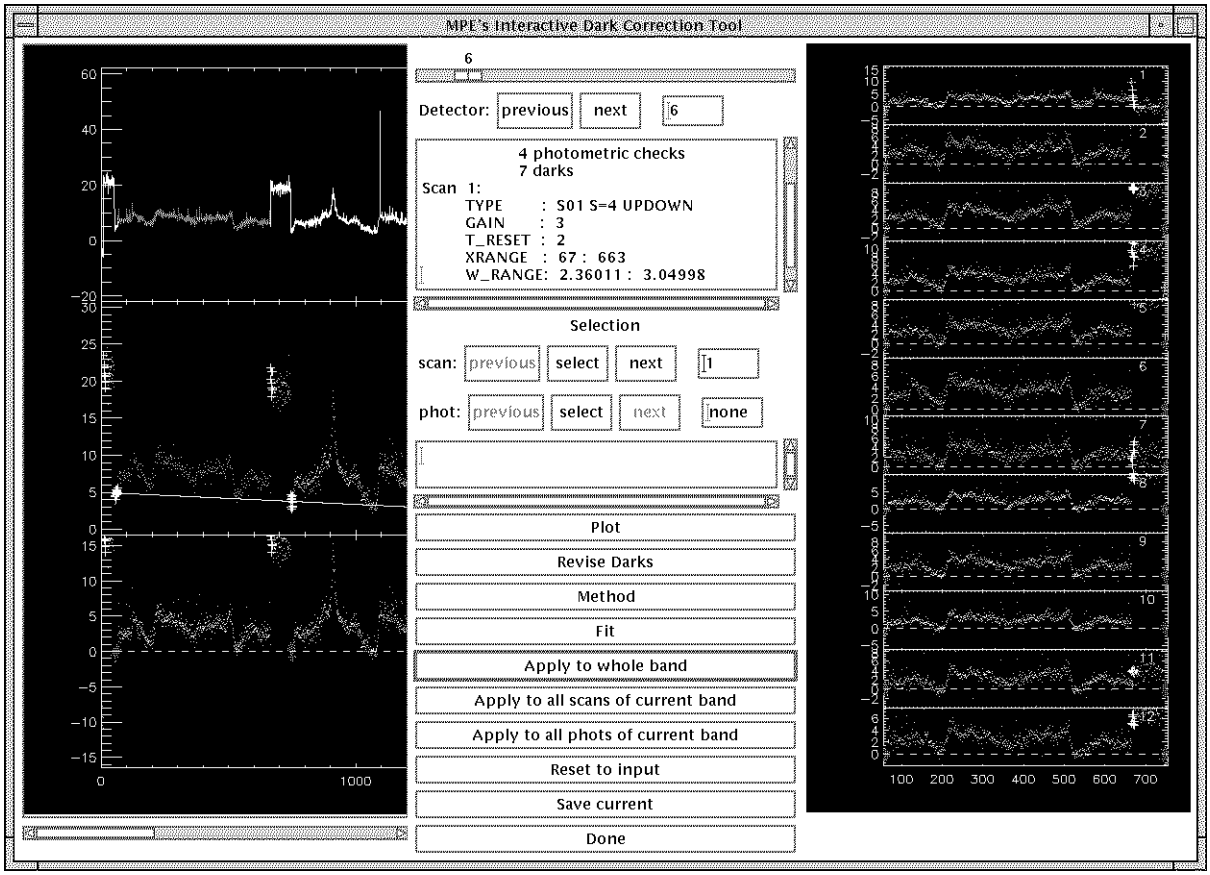

Figure 9: The DARK_INTER main window for an SWS01 observation after dark subtraction for scan 1 of band 1. As scan 1 in this specific observation has no next dark, the previous dark of the next scan was selected with the [cursor] button of the 'method' window to perform linear interpolation.

12.3 Saving results

The current status of processing can be saved with the [save current] button of the DARK_INTER control panel. After clicking this button, the 'save_spd' window pops up. Three different save formats are offered: IDL saveset, IDL save with **/xdr** option and standard fits format. The default file name is DARK_INTER_SAVE.XDR. This file contains the current output SPD as it would be created on exit of DARK_INTER now. In case of a crash, this file can be restored and used as starting point for a new DARK_INTER session. The name of the restored variable will be OUTSPD. To resume a session, just enter: *spd_D* **= dark_inter(restore,'***dark_inter_save.xdr***',** *sctbl*).

The output SPD contains the latest dark subtracted flux values for all scans and photometric checks, which have been processed in the DARK_INTER session. These scans and photometric checks should not be touched in further DARK_INTER sessions since dark subtraction of such scans and photometric checks cannot be improved later on. All scans and photometric checks which have been processed automatically (= blindly!) via [apply to whole band] and similar buttons, should be checked very carefully before saving intermediate data products. The result of dark subtraction with DARK_INTER can be checked with tools like MASK_INTER, PATCH_SPD and PLOT_PHOT HowTo process SWS data from SPD to AAR.

The [done] button closes the DARK_INTER main window (and, if still open, also the 'plot', 'revise darks' and 'method' windows) and it creates and returns the output SPD to the IDL session. **Make sure to save the output SPD immediately after completing DARK_INTER!**

13 HowTo make interactive updown correction

This document[11] describes how to perform interactive updown correction with the OSIA routines UPDOWN_INTER, UPDOWN1_INTER, and PATCH_SPD. These routines use the updown structure of SWS01/02/06 and SWS07-SW scans to determine a pragmatic correction for memory effects, updown effects and for slopes remaining from imperfect dark subtraction. The importance of these effects depend on the flux level. Detector memory effects have a strong impact on data in case of medium and high flux levels (via dark signal jumps) or strong flux changes (signal to signal variations), especially in band 2 and 4. The other effects are more important at low flux levels.

Whenever up and down scan show significant systematic differences, updown correction is a necessary step in the reduction of SWS data. Determining the cause of such differences facilitates their removal. As the mode of scanning varies between AOTs it was necessary to develop AOT tools specific to each mode.

13.1 Introduction

The OSIA routines UPDOWN_INTER, UPDOWN1_INTER, and PATCH_SPD are interactive modules to perform high quality updown corrections. They are used instead of the pipeline module UPDOWN (*see the dc-document* updown) which is usually executed between the dark subtraction and responsivity calibration (*please refer to* HowTo process SWS data from SPD to AAR). The pipeline routine UPDOWN was thought to correct for detector memory effects. However, in the absence of a suitable algorithm, the pipeline currently passes the data to the next step without any 'up-down' correction being made. Therefore, at present, the interactive tools are the only means of including updown corrections into the analysis of SWS data. This document gives detailed instructions how to use these tools.

A key issue is to understand how the interactive updown correction tools work. With the exception of SWS07-FP scans, all SWS observations are taken as pairs of **up** and subsequent **down** scans[12]. For each individual detector these scan pairs are performed in exactly the same way and reduced with exactly the same calibration, only the scan direction is different. **However**, the shapes of the corresponding up and down scans may be different caused by, for example, memory effects. In the interactive updown correction the difference between the scans is used to correct the slopes such that both scans match. Loosely speaking, the up and down part of each scan pair are flatfielded onto themselves per detector.

As the SWS scanning strategy depends on the individual scan mode, there are different tools for updown correction for the three SWS grating AOTs, which are described in the following three sections.

13.2 HowTo make updown correction for SWS01 observations

***** WARNING *** : It has not been possible to fully implement this module in the current version of OSIA therefore the recommended procedure for SWS01 observations is not to perform any up-down corrections on the data but to pass to the next stage of data processing (*refer to* HowTo process SWS data from SPD to AAR) or use PATCH_SPD (*see the dc-document* patch_spd).**

UPDOWN1_INTER is an interactive tool to make updown corrections for SWS01 observations. Rather than implementing detailed models for detector memory and updown effects, it tackles the problem by deriving a slowly varying correction which, when applied, makes the up and down scans agree. UPDOWN1_INTER is called as an IDL function and has three arguments, *spd*, *sctbl* and *aotlr*.[13] These arguments are idl variables containing the input SPD (after dark subtraction), its scan table structure and the corresponding SWS01 grating position information. To call UPDOWN1_INTER, enter:

spd_U = **updown1_inter**(*spd, sctbl, aotlr*)

The scan table *sctbl* should be derived from the SPD and be printed in advance with the OSIA commands MAKE_SCTBL

[11] This document is based on notes and recipes by Dietmar Kunze, Dieter Lutz, Eckhard Sturm and Michele Thornley.
[12] Nore: Up and down refer to grating scanner position corresponding to a 'down' and 'up' scan in wavelength.
[13] The names for the *variables* are only representative and can, of course, be chosen by the user.

and PRINT_SCTBL (*please refer to* HowTo process SWS data from SPD to AAR). The *aotlr* file is created as an optional by-product during the derivation of the SPD from the ERD: *spd* = **dspd**(*erd, aotlr=aotlr*). Therefore detailed analysis of SWS01 data has to start from ERD level (*please refer to* HowTo process SWS data from ERD to SPD). If the names of the in- and output SPD are different an output SPD with the specified name *spd_U* is created, otherwise the input SPD is overwritten.

WARNING: The **dspd** command to derive a SPD from an ERD is not available in OSIA V 1.0!

The *aotlr* file is used to compute the effective grating positions which are necessary to compute the proper difference between the SWS01 up and down scans. The grating information given in the SPD itself is not sufficient for this because the SPD holds only the grating positions at the start of each reset interval. As the scanner for SWS01 observations moves during the reset interval, the scanner positions provided by the SPD for corresponding points of the up and down scans are different. The array of effective grating positions *effpos* is set up for all scans in all bands immediately after the call of UPDOWN1_INTER. Its computation takes quite some time (have a coffee!). If the optional parameter *effpos=effpos* is added to the parameter list of UPDOWN1_INTER, the *effpos* array is returned and can be used if this routine is called for a second time.

After calling UPDOWN1_INTER the 'interactive AOT1 updown correction' window pops up (*see fig. 10*). It consists of a multiple plot window with a control board on the right side. Until the computation of the *effpos* array is finished, the message "creating array effpos – please wait !!!" is displayed. Afterwards the number of valid scans present for the current band is shown.

Updown correction is performed band-by-band. The current band is defined by the selected detector. The default is detector and band number 1. An individual detector can be selected with the detector slider at the top of the control board. The corresponding band is displayed in the band number entry field. If the band number is changed or the [previous] or [next] (band) button is used, a new band is selected (NOTE: any entry of a valid band number [1,..,4] must be confirmed by pressing the [RETURN] key). At the same time the first detector of this band is selected and the slider is moved correspondingly.

The number of the currently selected scan is given in the scan number entry field. The default is scan number 1. The selection can be changed with the [previous] or [next] (scan) buttons or by entering a new scan number (NOTE: any entry of a valid scan number must be confirmed by pressing the [RETURN] key). UPDOWN1_INTER considers only the science scans with non-zero wavelength ranges (compare the sequence of scans and their wavelength ranges in the scan table!).

The upper plot on the left side of the UPDOWN1_INTER window holds a stripchart of the data which contains the currently selected scan. The up and down scans are displayed in yellow and green respectively. The second plot shows the up and down scan overplotted onto each other and the third plot either the ratio or difference between these scans depending on whether the 'gain' or 'offset' mode is selected. These plots have the same x-scale (= effective grating position). Note that the wavelength in the plots *decreases* to the right.

The settings in the middle part of the control board determine how the actual updown correction is performed. The 'up' and 'both' buttons define whether *only the up* scan or *both* scans are corrected. If 'up' is selected then the up scan is corrected to match the down scan. This implies that the large scale structure of the result will be that of the down scan (which for high fluxes is assumed to be less affected by memory effects). If 'both' is selected, up and down scan are corrected to match each other. This option is preferable if strong signal variations (e.g. due to RSRF features) are present and both scans are assumed to be affected by memory effects in a similar (but opposite) way.

The 'offset' and 'gain' options select whether the corrections are determined for additive or multiplicative effects. For flux levels clearly below 100 [Jy], the 'offset' mode is recommended. The 'gain' option should only be used for observations with strong signal, where uncertainties due to dark subtraction can be neglected with respect to gain effects.

If 'both' scans are to be corrected, half of the 'offset' mode correction is is applied to each of the scans while for the 'gain' mode the square root of the correction is applied to each of the scans. If only 'up' scans are to be corrected the full correction is applied to the up scan alone.

The 'exclude' and 'include' options determine whether bad data points, such as masked points or points flagged as glitched

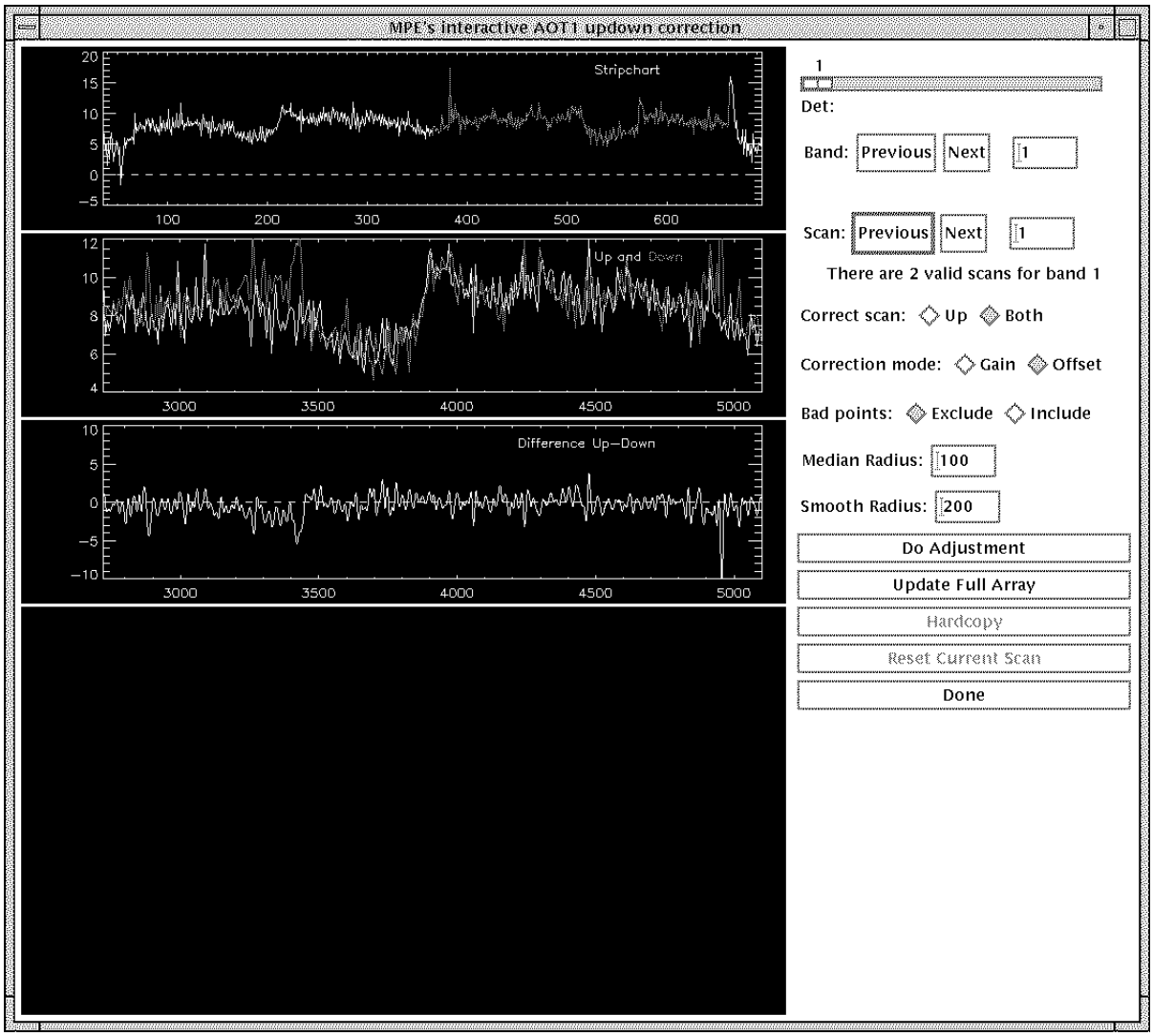

Figure 10: The UPDOWN1_INTER window for an SWS01 observation after setup.

Band	Scan	R(median)	R(smooth)	method
1	1	50-100	50-200	offset (or gain)
1	2	”	”	”
2	1	”	”	”
2	2	”	”	”
3	1	”	”	offset (or gain)
3	2	100-200	200-400	”
3	3	”	”	”
4	1	”	”	”

Table 14: Recommended settings for UPDOWN1_INTER.

or nodata, are ignored during the computation of the correction or not. Masking with MASK_INTER should be done in advance, if large peculiar differences appear between up and down scan, e.g. due to signal jumps or spikes with long tails. If not masked, such peculiarities could dominate the difference and also the final result.

Since memory effects often induce deviations in SWS01 updown scans, which cannot easily be fitted by simple functions, a smoothing approach was chosen in UPDOWN1_INTER. The ratio or difference between up and down scans is smoothed with a median filter followed by simple boxcar smoothing to eliminate distortions by single outlying points. The median and smooth radius can be specified (NOTE: any entry of a median or smooth radius must be confirmed by pressing the [RETURN] key). General recommendations for the radii are listed in table 14. The smallest values are appropriate for the slowest AOT SWS01 with speed 4. Whereas, larger values must be used for the faster modes with speed 3 to 1. The smoothed difference or ratio is then used to correct the up scan accordingly to match the down scan or to correct both up and down scans to match each other.

The effect of updown correction for the current scan and detector is displayed if the [do adjustment] button is pressed. Then the smoothed fit to the difference or, in the case of 'gain', the smoothed fit to the ratio and the corrected scans are overplotted. The corrected up scan is displayed in red. If 'both' is selected, additionally the corrected down scan is shown in blue. UPDOWN1_INTER performs at best if the fit to the difference/ratio represents the fitted data well (i.e. neither does it follow the noise too closely nor introduce artifacts into the spectrum) and if the fit is not driven by peculiar features in the individual scans. Iterate the radii until a satisfactory solution is found.

Click on the [update full array] button to actually perform the updown correction for the current scan. The ratios and fits for all 12 detectors of the current band are shown in the lower left panel. As the updown correction in UPDOWN1_INTER is always performed on the input flux, alternative correction modes, radii etc., can be tried and tested. The final method applied determines the output. If no real improvement can be found use the [reset current scan] button to reset the output data for this scan.

All non-zero science scans of band 1 to 4 should be treated. For very faint sources UPDOWN1_INTER should *not* be used to correct band 4 (= LW2) since glitch tails usually dominate the solution.

A general recipe, how to use UPDOWN1_INTER, looks like:

- select the first detector of a band
- perform updown correction scan-by-scan:
  - select the first/next scan
  - select correction mode: 'both' scans (or the 'up' scan), 'offset' (or 'gain') mode, 'exclude' (or 'include') and the proper median and smoothing radii
  - click [do adjustment] to display the effect for the selected detector
  - then press [update the full array] to actually perform the correction on all detectors in the band for the current scan and check the result

  - repeat this procedure for all scans of the current band
- repeat this procedure for all grating bands

A hardcopy of the plot panel can be made with the [hardcopy] button. A postscript file, with the default name *idl.ps*, is created which can be sent to the printer with standard printing commands. Print this file immediately as it will otherwise be overwritten. The [done] button closes the UPDOWN1_INTER window and creates the output SPD.

WARNING: using UPDOWN1_INTER requires familiarity with SWS observing modes and detector behaviour. If the tool is used in the wrong way it can introduce weird effects or as the author of UPDOWN1_INTER says: *"you can do terrible things to your data, if you apply this tool wrongly!"*

13.3 HowTo make updown correction for SWS02 observations

UPDOWN_INTER is a tool to make interactive updown corrections for SWS02 observations. The core idea is to fit some simple functions, which describe the detector memory effects reasonably well, to the difference or ratio of up and down scans and correct the data appropriately. Correcting with a linear fit also proved to be useful to remove slopes remaining from imperfect dark subtraction. UPDOWN_INTER is an IDL function with two arguments: *spd* and *sctbl*, which are the names the IDL variables containing the input SPD (after dark subtraction) and its scan table structure[14]. To call UPDOWN_INTER, enter:

spd_U = **updown_inter**(*spd, sctbl*)

The scan table *sctbl* should be derived from the SPD and be printed in advance with the OSIA commands MAKE_SCTBL and PRINT_SCTBL (*please refer to* HowTo process SWS data from SPD to AAR). If the names of in- and output SPD are different, an output SPD with the specified name *spd_U* is created, otherwise the input SPD is overwritten.

After calling UPDOWN_INTER the 'interactive AOT2 up-down correction window' pops up (*see fig. 11*). It consists of three panels: a control board in the middle and two multiple plot windows on each side.

The Updown correction is performed band-by-band. The current band is defined by the selected detector. The default is detector and band number 1. An individual detector may be selected with the detector slider at the top of the control board. The currently selected band is displayed in the band number entry field. If the band number is changed or the [previous] or [next] (band) button is used, a new band is selected (NOTE: any entry of a valid band number [1,..,4] must be confirmed by pressing the [RETURN] key). At the same time the first detector of this band is selected and the detector slider moves correspondingly.

The text line between the detector slider and band selection states how many scans are present for the current detector. UPDOWN_INTER counts only science scans with non-zero wavelength ranges (compare the sequence of scans and their wavelength ranges in the scan table!). The first step after calling UPDOWN_INTER is to select a proper scan. If the default scan number is -1 this means band 1 does not contain not any science scans. To select a scan enter the desired number in the scan number entry field (NOTE: any entry of a valid scan number must be confirmed by pressing the [RETURN] key). Usually one starts with scan number 1. The scan selection can be changed with the [previous] and [next] (scan) buttons or by entering a new scan number. The 'start index' of the selected scan in the SPD, the 'scan length' in reset intervals and the 'scan step' width, which is the index increment per reset interval in the SPD, are displayed in the control panel.

As soon as a scan has been selected, the four plots in the left panel are updated. The upper plot shows a stripchart of the flux of the selected detector plotted against SPD index. The displayed range is larger then the index range of the selected scan. The second plot displays the up and down part of the selected scan overplotted on a wavelength scale. The other two plots show the difference or ratio between the up and down scan and the standard deviation of this difference or ratio plotted against a common x-scale.

Before the updown correction is done, the appropriate correction method must be selected from the middle part of the control board. Two basic correction methods are offered:

[14] The names for the *variables* are only representative and can, of course, be chosen by the user.

- in the 'fit' method a straightforward least squares fit is applied to the difference or ratio between the up and down scan. This fit is used to correct either only the up scan or both scans such that their large-scale behaviour becomes the same.
- in the 'direct' method a correction is directly applied to the complete updown scan. Together with the up/down matching condition for the true spectrum this method is the solution of a minimization problem which is additionally constrained by the requirement that the turning point remains unchanged.

The 'fit' method is usually used. The 'up' and 'both' buttons then define if the correction is applied only to the up scan or to both scans. The first option is especially suitable to tackle memory effects from large dark signal jumps, while the second option is more robust[15] and preferable if strong signal variations are present or if up and down scans differ due to a residual linear dark current drift.

The 'offset' and 'gain' options select whether the corrections are determined for additive or multiplicative effects. For flux levels clearly below 100 [Jy], the 'offset' mode is recommended. The 'gain' option should only be used for observations with strong signal, where uncertainties due to dark subtraction can be neglected with respect to gain effects.

If 'both' scans are to be corrected, half of the 'offset' mode correction is is applied to each of the scans while for the 'gain' mode the square root of the correction is applied to each of the scans. If only 'up' scans are to be corrected the full correction is applied to the up scan alone.

The 'exclude' and 'include' options determine whether bad data points, such as masked points or points flagged as glitched or nodata, are ignored during the computation of the correction or not[16]. Masking with `MASK_INTER` should be done in advance if large, peculiar differences appear between up and down scans e.g. due to signal jumps or spikes with long tails. If not masked, such peculiarities could dominate the difference and also the final result.

Different functions are available in a pull-down menu to fit the difference or ratio. The ones mainly used for SWS02 observations are the 'linear ramp' (which is the default) and the (single or) double exponential in case of dark signal jumps. Additionally there are the "Fouks" (1994, SPIE, 283, 2268) and "PHT-IST26" functions. They are found to fit the transient behaviour of infrared detectors. However they are not perfectly tuned to the SWS detectors and make no claim on reproducing a physical model for them. The "PHT-IST26" function sometimes works quite well. It is the only method which can fit a response function that contains a hook-like structure. A kappa-sigma clipping is performed before the fitting and the kappa value must be specified in the adjacent box (NOTE: any entry of a valid kappa [>2,..,100] must be confirmed by pressing the [RETURN] key). A typically value for kappa is 3.

The effect of the updown correction for the current scan and detector is displayed, if the [do fit] button is pressed. Then the fit to the difference, or in case of 'gain', the fit to the ratio and the corrected scans are overplotted in the middle plots of the left panel. The corrected up scan is displayed in red. If 'both' is selected, additionally the corrected down scan is shown in green. `UPDOWN_INTER` performs well if the fit to the difference/ratio represents the fitted data well and if the fit is not driven by peculiar features in the individual scans.

After a satisfactory solution is found, click on the [update full array] button to actually perform the updown correction for the current scan. The ratios and fits for all 12 detectors of the current band are shown in the right panel. If the 'direct' method has been selected, plots of the ratios before and after the correction are displayed in the right panel. As the updown correction in `UPDOWN_INTER` is always performed on the input flux, alternative correction modes, fit functions etc. can be tried and tested. The final method applied determines the output. If no real improvement can be found, press the [reset to input] button to reset the output data for this scan.

All non-zero science scans of band 1 to 4 should be treated. For very faint sources, `UPDOWN_INTER` should *not* be used to correct band 4 (= LW2) since, in this band, glitch tails usually dominate the solution.

A general recipe of how to use `UPDOWN_INTER`looks like:

- select the first detector of a band

[15] The 'up' option may cause in some situations a sloped baseline in the final product.
[16] If most of the data are 'bad' one may get weird results with 'exclude'.

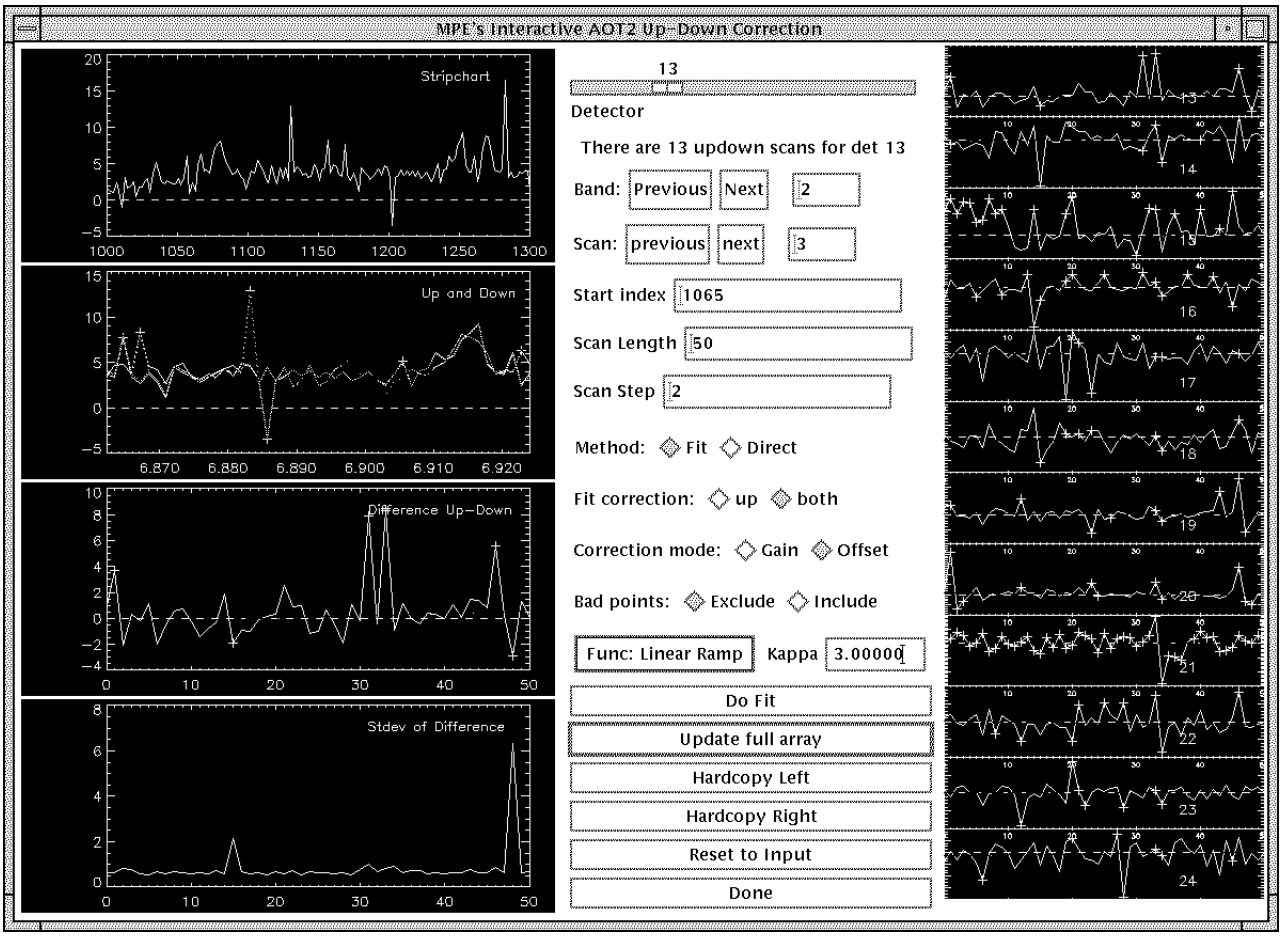

Figure 11: The UPDOWN_INTER window for an SWS02 observation after updating the full array.

- perform updown correction scan-by-scan:
  - select the first/next scan
  - select 'fit' method, correct 'both' scans (or the 'up' scan) in 'offset' (or 'gain') mode with 'kappa: 3' and 'linear ramp' (or 'double exponential') as fit function
  - press [fit] to display the effect for the selected detector
  - then press [update the full array] to actually perform the correction and check the result carefully
  - repeat this procedure for all scans
- repeat this procedure for all grating bands

Hardcopies of the plot panels can be made with the [hardcopy left] and [hardcopy right] buttons. A postscript file with the default name *idl.ps* is created whenever one of these buttons is pressed. Use standard print commands to send this file to the printer. Print the ps file immediately as it will be overwritten. The [done] button closes the UPDOWN_INTER window and creates the output SPD.

WARNING: using UPDOWN_INTER requires familiarity with SWS observing modes and detector behaviour. If the tool is used in the wrong way it can introduce weird effects or as the author of UPDOWN_INTER says: *"you can do terrible things to your data, if you apply this tool wrongly!"*

13.4 HowTo make updown correction for SWS06/07-SW observations

There is no dedicated tool for updown correction of SWS06 and SWS07-SW observations. These AOTs show a more complex scanning scheme than SWS01 and SWS02. For SWS06 observations up and down scans are separate scans with some non-science data points in between. Each up and down scan may consist of separate parts and may (depending on their length) be enclosed and even be interrupted by reference scans[17]. Nonetheless, the entire SWS06 scan, which is enclosed by the dark measurements, is symmetric. SWS07-SW scans show in principle also a SWS06-like scanning scheme but due to necessities of the parallel FP observations (e.g. to have a dark measurement at a certain time), they can be additionally interrupted. In such cases they may loose their symmetry.

As long as SWS06 and SWS07-SW scans are symmetric, their processing with interactive OSIA tools like DARK_INTER, MASK_INTER, etc., can be made much simpler if a pseudo-SWS02 scan table is used (*please refer to* HowTo process SWS data from SPD to AAR). Then, formally, the up and down parts of these scans are combined to single updown scan. Any intermediate non-science data are removed with a routine which is created as a by-product of the pseudo-SWS02 scan table. By using such a scan table UPDOWN_INTER can also be used for SWS06 observations. However, since UPDOWN_INTER is designed for short single line scans it is only applicable if the SWS06 scans cover a similar range as SWS02 scans and show similar, simple updown effects. A typical example are broad lines, which are scanned with SWS06, because the fixed SWS02 scan range is too small to get a proper baseline.

If a larger part of (or even a whole) an AOT band is scanned with SWS06 or SWS07-SW PATCH_SPD can be used for updown correction. This tools contains the functionalities of UPDOWN_INTER. In addition, different correction methods or fit functions can be applied to different parts of the scan. Furthermore, higher order polynomials are available as fit functions beside the linear ramp and single/double exponential. However, due to the single scan/single detector philosophy of PATCH_SPD there is no button to update the full array. Updown correction for a scan has therefore to be applied to all detectors of the band individually. See the preceeding section on UPDOWN_INTER and the online documentation of the tools and for more information please consult (*see the dc-document* updown_inter, patch_spd).

[17] Reference scans are short up scans around a reference wavelength specified by the proposer.

14 HowTo do RSRF correction interactively to minimise fringe residuals

14.1 Introduction

This HowTo document describes the means in OSIA to interactively correct for the instrumental response using the Relative Spectral Response Function (RSRF). The RSRF is the normalised response of the instrument.

A description of the RSRF's and their application is given in the SWS Handbook
(*see the following URL* http://www.iso.vilspa.esa.es/manuals/HANDBOOK/sws_hb/).

14.2 RSRF Features

The main reason to deviate from the use of the Pipeline implementation of the RSRF (RESPCAL in IA^3) is the presence of instrumental fringes. In particular for SWS band 3 the fringes are very strong and the standard correction as is done in RESPCAL is insufficient.

In OSIA a few routines have been developed to minimise the residuals of the fringes and/or to remove the residual fringes.

Besides fringes other instrumental artifacts can be present which are not properly corrected for by the RSRF. The SWS Handbook describes the present knowledge on these and instructs you what you should be aware of.

14.2.1 Observed Fringes

The spectral response as described in the RSRF and the response actually observed will in almost all cases differ. E.g. the fringes in band 3 as they are observed depend on a few factors:

Wavelength Calibration Even very small deviations from a perfect wavelength calibration can result in an offset of the observed fringes and the fringes in the RSRF. This offset can vary with wavelength.

Orientation in the SWS Aperture The instrumental response from a signal at a position offset from the center of the SWS aperture differs from that of a signal from a source at the aperture center. The wavelength calibration is affected and it affects the amplitude and details in the fringes (in particular in band 3A).

Extent or Source Structure The extent or structure of a source affects the observed fringes. Most noticeably it reduces the resolution of the observed fringes and possibly changes details in the fringes. Basically the response seen from a non-point source is the integrated response of many signals at different positions in the aperture.

14.2.2 RSRF Fringes

The fringes as they are described in the RSRF's have been derived from laboratory measurements using fully extended black-body sources. This limits the accuracy of the fringe description in the RSRF since the appearance of the fringes depend on observed signal, see the previous section (Observed Fringes).

It also means that the laboratory RSRF functions will never fully be able to reproduce the observed fringes since the basic information, the response from a point-source signal anywhere in the aperture, is lost.

An effort is being made to come to a better description of the RSRF using in orbit observations. This applies both to a more detailed description of the full RSRF from a point-source and an attempt to understand the impact of the orientation of the signal on the appearance of the fringes.

Besides the empirical approach there is also an attempt to come to a 'full' theoretical description of the fringes which can help to improve the understanding and correction of the observed fringes.

14.3 RSRF correction in OSIA

In OSIA the routine RESP_INTER has been designed to help minimise the residuals of the fringes after application of the RSRF in those cases where the standard RESPCAL fails. To do this it adapts the RSRF to match the data by changing the wavelengths of the RSRF (by means of a shift determined from a correlation with the observed fringes) and changing the amplitude of the fringes in the RSRF (either decreasing them or increasing them to match the amplitude of the observed fringes).

In some cases this gives very good results and the fringe residuals can be reduced to only a few percent. However in other cases this is still insufficient, although it gives a significant improvement compared to the standard Pipeline product. In those cases OSIA offers the possibility to further reduce the fringe residuals using FRINGES (which uses cosine fitting in wavenumber space) or AARFRINGE (using a similar method or using a Fourier Transform method).

14.4 RESP_INTER

RESP_INTER is used instead of RESPCAL. I.e. it is used after dark current subtraction and before the absolute flux calibration and velocity correction.

14.4.1 General use

In general it is advised to use RESP_INTER in interactive mode only for these bands where one has problems with the fringes. Normally the author himself will first use the standard RESPCAL to process all data. After that he uses RESP_INTER only for the bands in which there are problems. These bands he will then extract from the AAR produced this way and, if a full AAR spectrum is required, merge this with the other bands extracted from the AAR produced using RESPCAL (*please refer to* HowTo reduce a grating AAR for a basic description of data selection and merging).

14.4.2 Implementation

The corrections applied to the RSRF consist of a combination of a shift and a smoothing or enhancing factor. The shift is determined from a correlation of the observed fringes with the fringes in the RSRF. The smoothing is applied by default for the fast speed AOT-1 (speed 1,2 and 3). For AOT-1 speed 4 and AOT-2 and AOT-6 the amplitude of the fringes is adapted via an amplification factor. Figures 12, 13 and 14 give an illustration of the plots and widgets RESP_INTER produces. Besides the plots RESP_INTER also gives the user command line info over the corrections calculated and applied.

The defaults can be overruled using the /ENHANCE or ENHANCE=0. The first will use an amplification factor in all cases (this can be an amplification smaller than 1, e.g. in the case of a fast speed AOT-1 observation), the last a smoothing in all cases. In case smoothing is selected and the amplitude of the RSRF fringes is smaller than the observed fringes, this will result in no smoothing and of course a non-optimal result.

The default operation mode of RESP_INTER is the interactive mode. In this mode the user is presented with all correlations and derived corrections. He can adapt and disable certain corrections which appear to be wrong and test the impact this has on the data.

An output parameter structure is created which can also be used on input. This makes it possible to e.g. derive the corrections from one observation and apply these to another observation. This may e.g. be useful when an observation has too many lines in the spectrum as to make a good correction possible. Note however that the SHIFT information may be incompatible between observations and that one has to restrict oneself only to smoothing or enhancing. In this case care has to be taken that the SPD is created using the latest most accurate wavelength calibration, in that case the shifts are often small and its effect is small. With an old incorrect wavelength calibration, the effect of the shift can have a severe impact.

```
SPD = RESP_INTER( SPD, .... , PARAM=PARAM_STRUCT, /NOSHIFT )
SPD = RESP_INTER( SPD, .... , PARAM=PARAM_STRUCT, /USE_PARAM, /NOSHIFT )
```

14.4.3 Particular cases

As with many reduction methods there are always particular cases for which the defaults do not work or do not work optimally. A few examples are given here, in the future these will be extended and illustrated.

14.4.3.1 Fast AOT-1 observations For some fast AOT-1 observations better results have been achieved using the /ENHANCE option instead of the default smoothing. This may depend on the observed target though.

For the AOT-1 speed 1 and 2 observations the sampling can become too low. In those cases it will become difficult to get a good correlation between the RSRF and the observed fringes. For these cases the option DET=0 may be a solution. Using this option the signals of all twelve detectors in one band will be combined and one average correction will be derived.

14.4.3.2 Low signal data For low signal data it can also become difficult to get a good fringe correlation. At the moment the only extra option in RESP_INTER is the DET=0 option, to combine the signal of all detectors.

```
SPD = RESP_INTER( SPD, .... , DET=0 }
```

At the time of issue of this document RESP_INTER is under revision to enable the option to combine different lines covering the same wavelength region and thus increasing the SNR.

It is also possible to use the parameter structure derived from another similar but stronger target as input to RESP_INTER.

```
SPD = RESP_INTER( SPD, .... , PARAM=PARAM_STRUCT, /USE_PARAM )
```

14.4.3.3 Confused spectral regions When part of a spectral region is confused in the sense that there are many lines of e.g. when a molecular band is present, RESP_INTER will become confused and will apply wrong corrections. In such a case it is possible to apply a single point correction over a wavelength region which is (almost) free from confusing lines.

```
SPD = RESP_INTER( SPD, ..... , XRANGE=[15,16], NBIN=1 )
```

One correction derived from the region of 15 to 16 microns is applied to the whole of band 3A. This is an example which has been applied to real data where the region below 15 microns was dominated by strong molecular bands of C_2H_2 and HCN. Similar applications have also successfully been used.

It is also possible to use the parameter structure derived from another similar target (e.g. both extended or both point-source) without molecular bands as input to RESP_INTER.

14.4.3.4 AOT-2 data At the moment the results with AOT-2 data are a mixed success. The problem with AOT-2 data is that the wavelength range of one scan is fairly small, sometimes so small that only part of a fringe period is sampled. This makes it difficult to derive correction factors. It is hoped that with the release of the next version of RESP_INTER this may improve when the signals of different scans covering different wavelength regions can be combined.

It is also possible to use the parameter structure derived from another similar target (e.g. both extended or both point-source) but from a AOT-6 observation covering a larger wavelength region as input to RESP_INTER to reduce the AOT-2 observation.

14.5 Future Developments

A few changes/improvements are foreseen for the next version of RESP_INTER:

```
- Automatic masking of lines
- Signal combination of different lines
- Signal combination of different scans
- Separation of shift and smooth/enhance determination
    currently this is done in the same loop
```

and the next version of this HowTo:

```
- more examples
- more illustrations
```

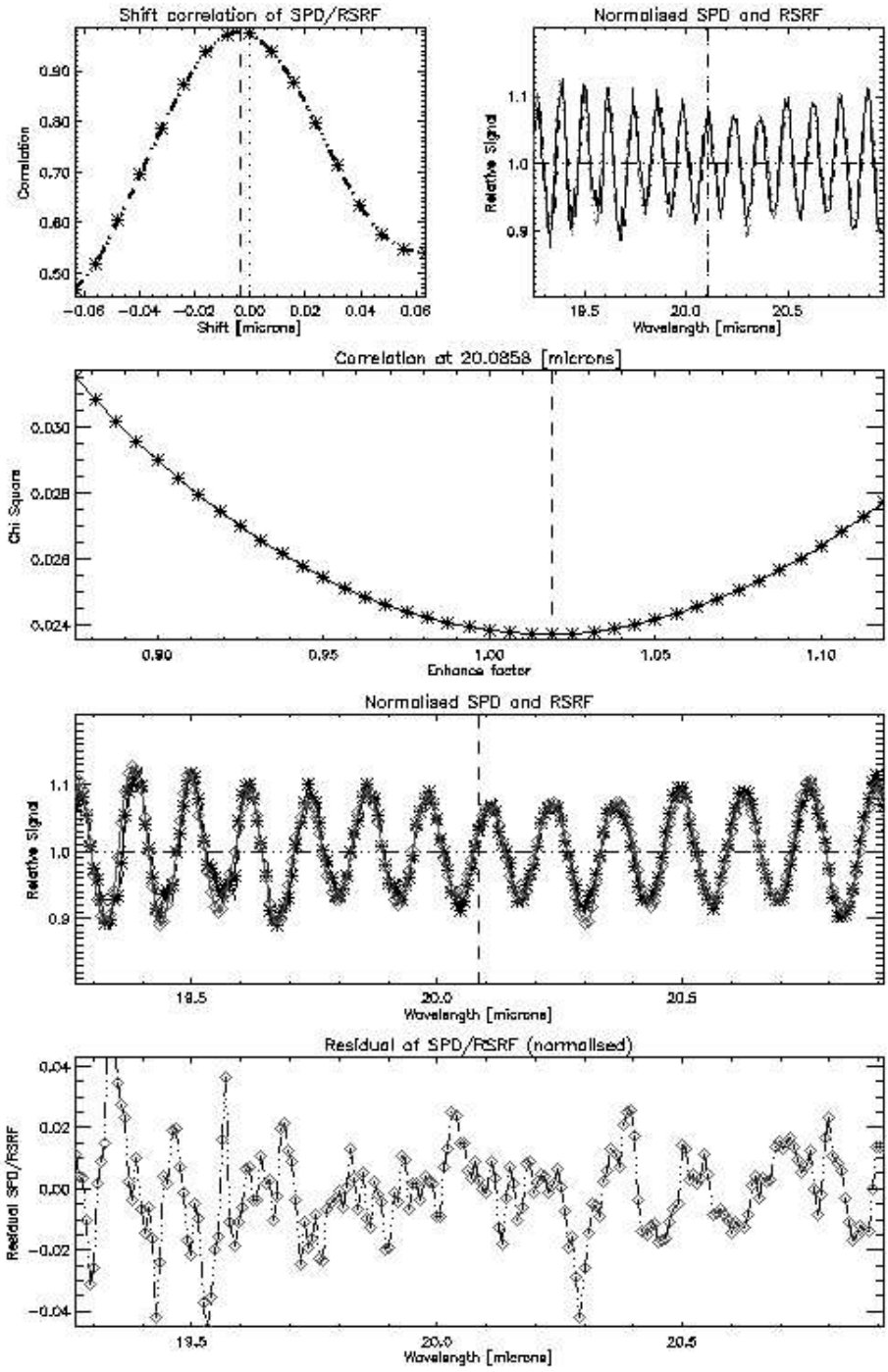

Figure 12: Determination of RSRF corrections

Figure 12 shows the corrections calculated by RESP_INTER. These corrections are calculated over a wavelength region around a number of wavelengths within one scan. Figure 13 shows the corrections applied over the full scan and figure 14 the widget with which the user can control the applied corrections.

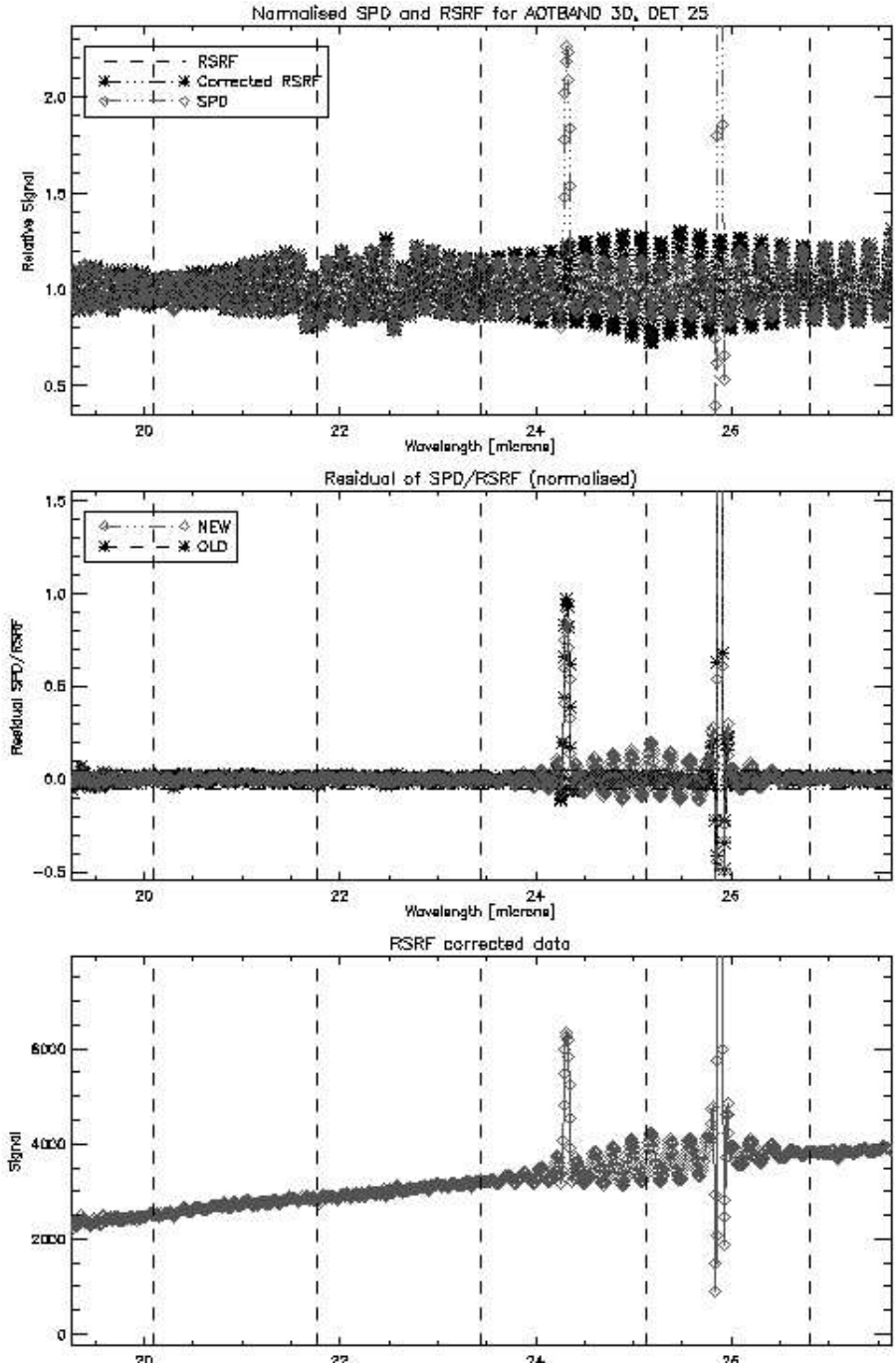

Figure 13: Calculated RSRF corrections applied to full scan

In figure 13 the calculated corrections (as illustrated in figure 12) are applied to the RSRF and the SPD data is corrected using this RSRF. In this particular case there is evidently a problem with the fourth correction (also see figure 14) as a result of the strong lines. This makes the RSRF correction in the second half of the scan quite poor. One of the planned updates for the next version of `RESP_INTER` is aimed to solve this problem. In the meantime the user can disable the correction via the widget shown in figure 14.

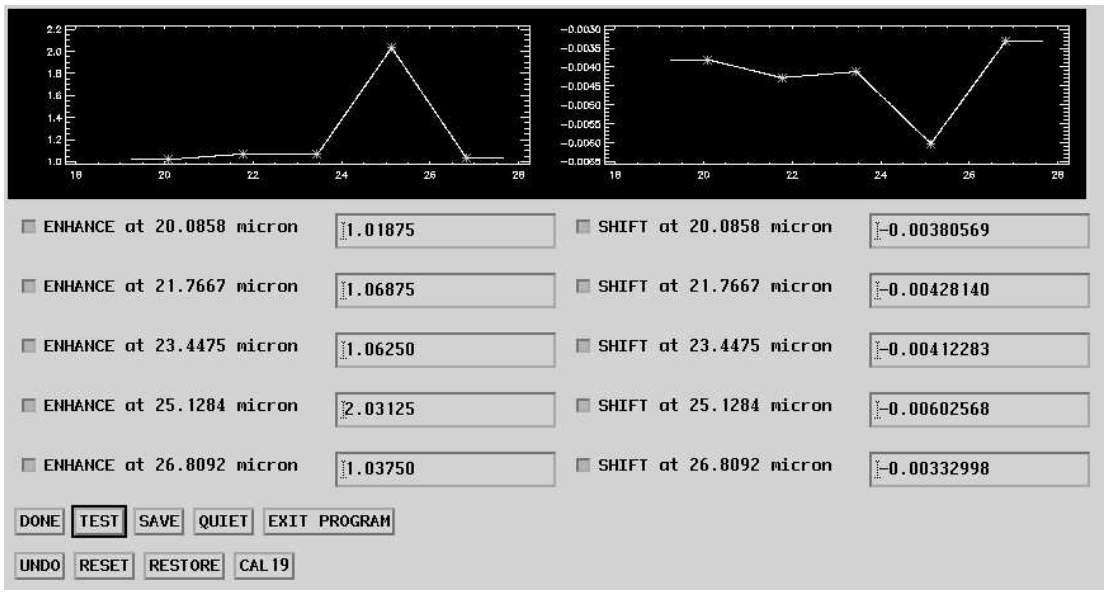

Figure 14: RESP_INTER control window

Figure 14 shows the control window of RESP_INTER. In this widget the user can control the corrections applied to the RSRF. In this case e.g. there is a clear problem with the fourth correction. The user can disable this one and push DONE to apply it and continue to the next scan or push TEST to first see the result.

15 HowTo minimise fringe residuals in SWS data

This HowTo describes how to minimise fringe residuals in SWS data. In all SWS AOT-bands fringes are present to some extent. The minimisation of fringes in OSIA is handled in two steps. The first step is to reduce the amplitude of the fringe residuals when applying the Relative Spectral Response Function (RSRF). The second step is to remove these residuals as much as possible.

15.1 Introduction

Fringes originate on plane parallel surfaces in the light path of the instrument. These surfaces act as Fabry-Pérot etalons each of which can add unique fringe components to the source signal. In the infrared wavelength range surfaces separated by a few mm up to a few cm form the most efficient FP's. It is therefore not surprising that fringes are seen at most wavelengths in the SWS instrument. The fringes originate from a number of sources in the instrument. These can be filters, plane mirrors or detector surfaces. The last is the most conspicuous source of fringing for the band 3 detectors which utilise BIBIB detectors.

While in principle the step `RESPCAL` of the pipeline is designed to remove RSRF features such as fringes, there are several reasons why in practice fringe residuals are left after applying this step using the standard CAL25 RSRF.

- Since the SWS spectral resolving power depends on source size, so does the RSRF. Point sources produce sharper and higher amplitude fringes than extended sources. The current RSRF is still fundamentally based on laboratory extended source data.
- For point-like sources, pointing shifts in dispersion direction and, at least for band 3, shifts orthogonal to the dispersion, change the RSRF and cause residual fringes. The same also applies to small wavelength calibration drifts.
- SWS01 introduces additional smoothing reducing the amplitude of the fringes.

For these reasons it will never be possible to derive a complete set of responsivity functions suitable for all observations. Therefore in OSIA routines are available to help minimise the fringe residuals in your observation as much as possible.

In most SWS AOT-bands fringes have been detected. In some cases they have only been seen in the RSRF at a very low level ($< 1\%$) in other cases they can only be detected in high resolution and high signal to noise observations and not in the lower resolution RSRF. Table 15 gives a quite complete overview of all the main fringe components and the amplitude and type of observation in which they have been detected. In the paper by Do Kester [3] a complete description of the fringes in SWS data and fringe models for these are described.

15.2 OSIA modules

In OSIA three main routines have been developed to help reduce and remove fringe residuals. One of these (`RESP_INTER`) replaces the pipeline RSRF correction step (`RESPCAL`). The other two (`FRINGES` and `AARFRINGE`) remove the remaining fringe residuals after RSRF correction.

`RESP_INTER` replaces the pipeline `RESPCAL` step and tries to tackle the problem by countering the basic reasons: the RSRF is shifted and smoothed/enhanced (see section 15.3).

`FRINGES` works on an SPD or AAR (standard or rebinned) **after** an RSRF correction with `RESPCAL` or `RESP_INTER` has been applied. It implements a robust method of iteratively fitting sine functions to the data and correcting for them (see section 15.4).

band	D mm	range of D mm	RSRF %	Residual %	AOT
1A	4.20	4.1-4.3	-	8	6,2
1B	4.30	4.2-4.4	-	3	6,2
1D	4.45	4.3-4.5	2	5	6,2
1E	4.58	4.4-4.6	-	1	6,2
2A	4.43	4.4-4.6	1	2	6,2
	0.89	0.8-0.9	0.5	-	
	2.39	2.3-2.4	0.3	-	
2B	4.50	4.4-4.7	2	1	6,2
	0.89	0.8-0.9	0.6	-	
2C	5.86	5.6-6.1	4	4	all
	0.89	0.8-0.9	2	-	
3A	3.37+3.25	3.2-3.4	25	5	all
	5.26+5.72	5.0-6.0	10	5	
3C	3.25+3.38	3.2-3.4	17	4	all
3D	3.25	3.2-3.4	17	2	all
3E	3.28	3.2-3.4	19	2	all
4	13.25	13.2-13.3	0.2	-	-
4A	5.27+5.73	5.0-6.0	10	3	all
4C	6.67	6.5-6.8	1	-	all
4D	-	-	-	-	-

Table 15: Main fringe components per AOT band. The second column gives the effective thickness for the main components. Sometimes there is more than one component. These are listed separately when the components are quite distinct. Column 4 and 5 list the compound amplitude in percent, of the fringe components in resp. the RSRF and of the residual fringes in some observations on point sources. The last column list the AOT numbers where these residual fringes might be found.

AARFRINGE works at the level of rebinned AAR's. This is not optimal and whenever possible the above routines should be used. In cases where these fail it may be possible to achieve good results using AARFRINGE (see section 15.5).

The user should always be aware of the fact that the defringing of SWS data remains an interactive process which should be done with care requiring the judgement and common sense of the user. If applied wrongly, all mentioned methods and algorithms will introduce artifacts into the data which can lead to an erroneous scientific interpretation.

'Good data' for which excellent results are obtained typically consist of much clean continuum plus some lines. Even then, a rule of thumb may be that defringing rarely reveals complete miracles - lines and features which after defringing are good enough for scientific analysis usually have been suggested to the educated eye already in the fringy data by irregularities in the fringe pattern.

More complex cases are data with molecular bands covering most of the data, in particular if their frequencies are commensurate with fringes. Other problems occur for SWS02 data with just 1 or 2 fringes and broadish lines that blend into the fringes.

In most cases, doubt or no doubt, it is strongly suggested to try different reductions to assess the robustness and credibility of the results. This may be trying different methods or modifying the reduction parameters to the routines. Some examples are given in the sections below, see for example section 15.3.2.2, 15.4.2.1 and 15.4.2.2.

15.3 RESP_INTER

RESP_INTER is written to replace RESPCAL. It will reduce the amplitude of the fringe residuals through modification of the RSRF by matching the fringes in the RSRF with the observed fringes. It will shift and smooth or enhance the RSRF.

It shifts the RSRF to compensate for wavelength shifts due to pointing offsets, source morphology or calibration drifts. The smoothing or enhancing is done to decrease or increase the amplitude of the RSRF fringes to compensate for the difference in resolution between the RSRF and the observational data.

RESP_INTER must be run after FLUXCON and before VELCOR. FLUXCON must be run first because the default is to derive single correction parameters for all detectors together. It is therefore important that the detector signals are flatfielded first to guarantee the most optimum result. This is different from the pipeline implementation where RESPCAL is executed first. Since the responsivity calibration and the absolute flux calibration are orthogonal operations their order can be reversed without any consequences.

In most cases there will still be small fringe residuals left after using RESP_INTER. This means that after RESP_INTER, extra processing using FRINGES is required. However, this does not mean that it can just as well be omitted. Experience has shown that the combination of the two routines in general gives best results. With RESP_INTER the remaining fringe residuals are better defined mostly because of the correction for wavelength shifts. Because of this FRINGES is able to find fringe residuals down to a much lower level.

15.3.1 Default operation

The defaults of RESP_INTER are chosen such that it is expected to produce good results for most data. By default RESP_INTER will:

1. Only modify the RSRF for bands 3 and off-band 4 data.
2. Smooth the RSRF for AOT-1 speed 1 and 2.
3. Enhance the fringe amplitude in the RSRF for AOT-1 speed 3 and 4 and for AOT-2 and 6.
4. Determine single correction parameters for the combined detectors. This gives stable results in all cases. For good signal to noise data a correction per detector may improve the results. ... try it and compare the results.
5. Fit the same continuum to RSRF and data. This means the same type of continuum fit over the same wavelength range.
6. Mask strong unresolved spectral features.
7. Correct per line and scan direction.

When the user does not specify any AOT bands all bands will be RSRF corrected. Only for all band 3 and the off-band 4A data RSRF modifications will be determined and applied. For all other bands RESPCAL corrected data is returned. When the user specifies particular AOT bands he should be aware that only those AOT bands will be RSRF corrected. This is to facilitate the possibility to make successive calls to RESP_INTER with different options per AOT band.

RESP_INTER plots the corrections and presents these to the user for inspection and approval. If the user is confident enough in the processing and knows his data to be without spectral features which are not masked properly he can tell RESP_INTER to continue QUIET-ly or he can process the observation with INTERACTIVE=0 and the /NOPLOT options. In this mode RESP_INTER can be used without problems for batch processing of e.g. large data sets.

15.3.2 Reduction examples

Below we give some examples of the way in which RESP_INTER can be used. For the explanation of the syntax you should consult the OSIA dc file.

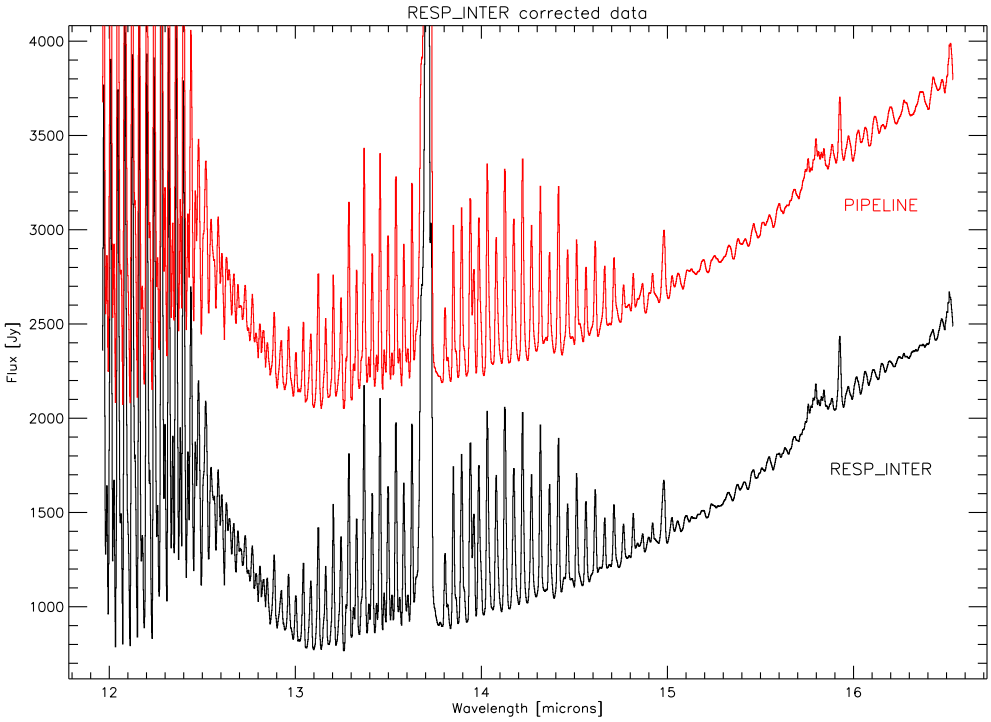

Figure 15: Band 3A data with dominant emission from several molecular bands complicating the correction of RESP_INTER and prohibiting a defringing using FRINGES. In this case a single correction is determined on the stretch of data between 15 and 16 μm which is not completely dominated by molecular emission. The result is a reduction of the fringe residuals by more than 50 % which leads to improved model fitting and better detection of weak spectral features. In black the RESP_INTER corrected data is plotted and in red the standard processed data using RESPCAL. The improvements can best be seen in the better underlying continuum of the C_2H_2 emission band between 13 and 15 μm and in the better defined C_4H_2 molecular emission band around 16 μm.

15.3.2.1 Standard calls These are few different ways in which RESP_INTER can be used.

```
spd = resp_inter( spd )                                    ; correct all AOT bands
spd = resp_inter( spd, aot_band = ['3a','3c'] )            ; correct only bands 3A and 3C
spd = resp_inter( spd, aot_band = '3a', det=indgen(12)+25 )  ; correct per detector
spd = resp_inter( spd, param=params )                      ; all bands, return parameters for inspection
```

15.3.2.2 Spectral confusion In some cases the major part of an AOT band is dominated by real spectral features, e.g. strong molecular bands of C_2H_2, HCN or CO_2 in band 3A. In that case it may be possible to derive a correction over a wavelength range not dominated by spectral features. Figure 15 shows an example where a single correction was determined between 15 and 16 μm and applied to the whole band.

```
spd = resp_inter( spd, aot_band = '3a', xrange = [15,16], nbin = 1 )
```

In this case it leads to a visibly better result reducing the amplitude of the fringe residuals by more than 50 %. This makes it possible to better constrain temperature and density profiles in the physical-chemical models and improve the detection of or fits to weak spectral features.

15.3.2.3 Using parameter input If in a case like the one described in section 15.3.2.2 there is no useful wavelength range to be found it is possible to give correction parameters on input. The simplest is to use a single fringe amplitude correction.

```
spd = resp_inter( spd, aot_band = '3a', enhance = 1.2 )
```

This will enhance the fringe amplitude in the RSRF for band 3a with a factor of 1.2. This is approximately the difference between the fringe amplitude of a fully extended source (RSRF) and a point source (most astronomical sources) for an AOT-2 or 6. Scientific analysis, e.g. improved and more consistent modeling of the data, should then prove the validity of this correction.

Another possibility is to use a source of similar extent to derive amplitude correction parameters and use these.

```
spd1 = resp_inter(spd1, aot_band='3a', /noshift, param=params )
spd2 = resp_inter(spd2, aot_band='3a', param=params, /use_params)
```

In this spd1 is the reference spd and spd2 the problem spd. `/NOSHIFT` is used since the shift is more observation dependent than the fringe amplitude. A small difference in pointing can result in significant differences in wavelength shifts (though small in terms of wavelength accuracy, ~ 0.1 resolution element). So to avoid making things worse it is better to leave the shift correction out.

15.4 FRINGES

The `OSIA` routine `FRINGES` models the residual fringes in the data after application of the RSRF by either `RESPCAL` or `RESP_INTER`. The latter is preferred since in general the results after `FRINGES` are then better than when using `RESPCAL`. This is a result of the fact that the fringe residuals are better defined after the corrections in `RESP_INTER`.

`FRINGES` approximates each residual fringe component with a sinusoidal function giving a total fringe model (see Kester [3] for the full details) I defined as:

$$I = \sum_{n=1}^{N} A_n cos(4\pi D_n K) + B_n sin(4\pi D_n K)$$

With D_n is the effective thickness of the etalon. A_n, B_n and D_n are the parameters to be estimated and K is the frequency in wavenumbers.

The basic steps involved in the processing in `FRINGES` are:

1. fit a smooth background and subtract this
2. mask out unresolved spectral features
3. fit N fringe components
4. make total fringe model
5. subtract the total fringe model
6. add background in again

The number of fringe components is determined either based on the calculated evidence or on the tolerance and number of fringes as defined by the user. The standard automated mode is using the calculated evidence. In this mode the procedure accepts a found fringe component and continues searching for a new component as long as the evidence is increasing. This is the advised mode for general use.

Figure 16 shows two defringing examples. The first is in band 1A where we see the fringes in the data for a point source but not in the (lower resolution) RSRF. What at first sight may appear as noise turns out to be mostly real fringes. The second plot shows band 3A where the fringes are clearly seen in the RSRF and are still present in the RSRF corrected data. In both plots the input data, corrected data and RSRF are shown in the top panel and in the lower panel the background subtracted data and the composite fringe model is shown.

In good quality data fringe residuals can be detected down to a level of 0.1 %.

15.4.1 Advantages over FFT

There are three main reasons why the procedure adopted in FRINGES is superior to the use of FFT's.

1. The wavelength of the fringe can be estimated much better than in an FFT where it is limited to the Nyquist frequency, which in turn is determined by the number of data points.
2. It works on unevenly spaced data just as well as on a regularly gridded data set. Even when there are large gaps in the data like e.g. in an AOT2, it works fine.
3. It automatically determines how many fringe components there are present, avoiding possible overfitting or underfitting.

15.4.2 Using FRINGES

Over time FRINGES has evolved significantly. As much as possible the knowledge we have about the fringes and the pit-falls of the data have been caught in the routine and in the employed defaults. This means that in general the default use of FRINGES is expected to give good and stable results.

FRINGES is called to run on separate bands (combining the signal of all detectors and all scans) or on individual detectors. The input to FRINGES can be either an SPD or an AAR. In most examples we therefore use the name AARSPD to indicate this.

Some standard processing examples:

```
aarspd = fringes( aarspd, band = '3a' )                    ; all of band 3A
aarspd = fringes( aarspd, band = 3 )                       ; all of band 3A, 3C, 3D and 3E
aarspd = fringes( aarspd, indgen(12)+25, aperlist = 1 )    ; individual detectors of band 3A
aarspd = fringes( aarspd, band = '3a', /updown )           ; all of 3A but up and down scans separate
```

The processing is in general best done per (AOT-)band. It appears that the fringe residuals of the different detectors in a band are so much similar that real gains are made by combining them.

This does of course not mean that there are no cases where a non-default use of FRINGES will not give better results. The user should certainly not be discouraged to try what may at first sight appear to be unconventional uses of the routine. This is usually the best way to learn something new about the problem.

Below we will try to give some examples of 'advanced' use of FRINGES.

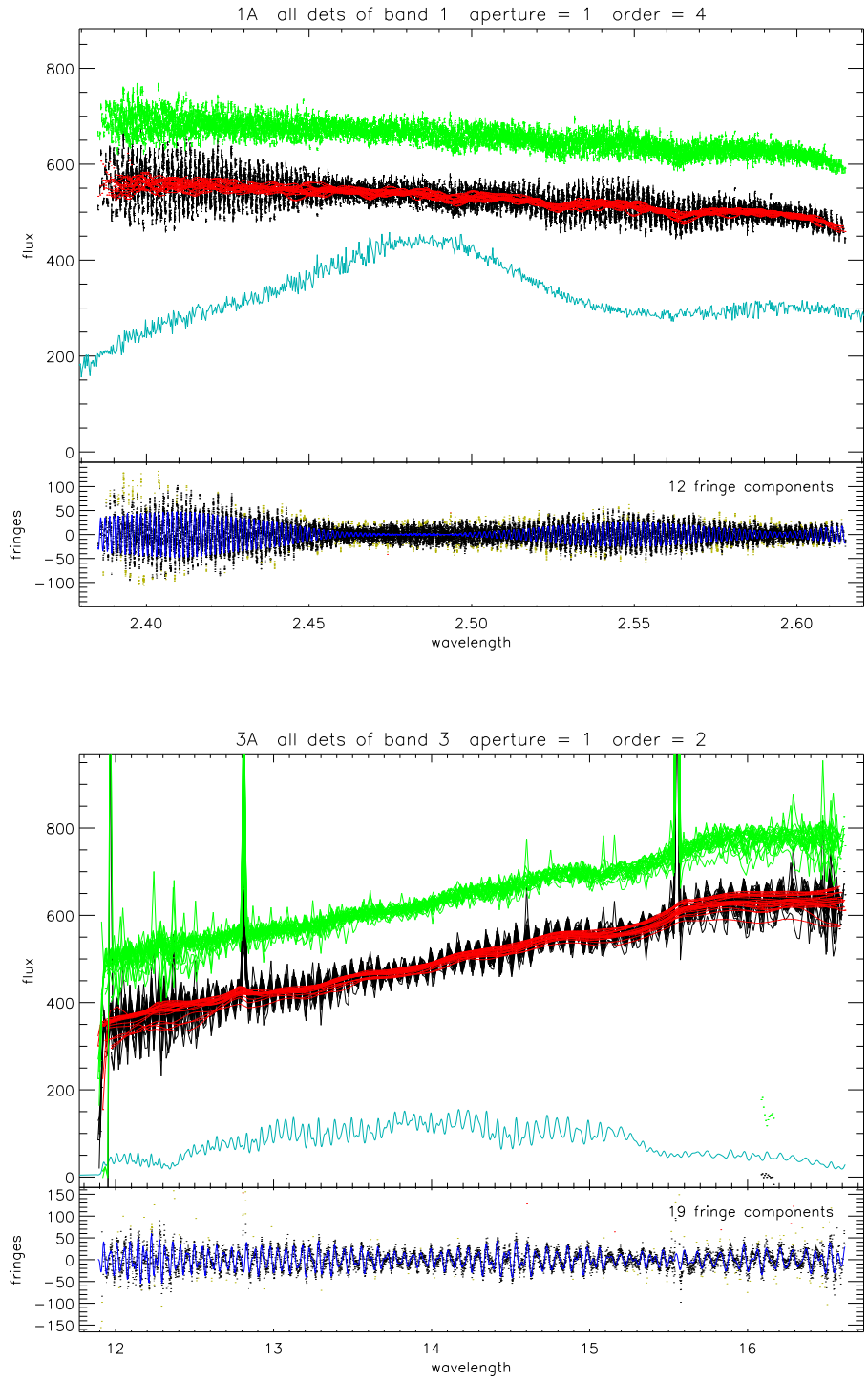

Figure 16: Defringing examples using FRINGES on band 1A and band 3A data. In the upper panel the data and the RSRF are shown. The input data is in black with the smooth background in red, the corrected data is plotted in green and is shifted for comparison and in light blue the RSRF (scaled to fit the plot). In the lower panel the background subtracted data is shown in black and the composite fringe model is overplotted in blue. Note that in band 1A none of the fringes are present in the RSRF while in band 3A the similarity between the fringes in the RSRF and the residual fringes is striking.

15.4.2.1 Up and down scans This is a test to decide which is the best defringing, a defringing with all data combined or with the up and down scans defringed separately.

```
aarspd1 = fringes( aarspd, band = '3A' )
aarspd2 = fringes( aarspd, band = '3A', /updown )
```

For this we use the evidence. The value in the last column of the IDL output of FRINGES 'LogPr' is the logarithm of the probability for that solution. When the up- and down-scans are reduced separately we may sum the probabilities. The difference of this with respect to the probability derived from the other processing gives us the evidence of one solution wrt. the other.

For the processing of the combined scans (to get aarspd1) we find a value of 2143.6. For the processing of the separate scans (to derive aarspd2) we find values of 1040.9 (UP) and 1027.2 (DOWN) adding up to 2068.1.

So the evidence of solution 1 with respect to solution 2 equals 10 * (LogPr1 - LogPr2) dB being 755 dB. We therefore decide that aarspd1 is to be preferred over aarspd2 by more than 750 dB. A significant difference. A visual check by overplotting both results will confirm this.

15.4.2.2 Limiting the cycle range In the case when e.g. molecular bands are present which have a beat frequency of the same order as the fringes we can do the following.

1. Reduce the SPD data using the pipeline modules ANTIMEM, DARK, FLUXCON. This is then optimally reduced with RESP_INTER and FRINGES.
2. Find the optimum reduction parameters for RESP_INTER and FRINGES by repeating the following until you are happy with the result.
   (a) run RESP_INTER with modified parameters, e.g. limited wavelength range or use less bins than normal, see section 15.3.2.2 for an example
   (b) run FRINGES modifying e.g. the cycle range (see table 16) and wavelength range
   (c) make a fit of the molecules to the reduced spectrum
   (d) make a synthetic spectrum over the same wavelength range as the data
   (e) judge the quality of the reduction by:
      i. estimating the amplitude of the fringe residuals in the data after defringing
      ii. defringing the CAL25 to get a second estimate of the amplitude of the fringe residuals
      iii. estimating the effect of the defringing on the synthetic spectra
         A. defringed with FRINGES
         B. molecular fit to defringed synthetic spectrum
      iv. make a scientific judgement on the results
3. Make a final fit to the spectra and use the results from 2-e in the error estimation of the fitted molecular parameters.

This procedure has actually been used with quite some success in a few scientific programmes. One example of this can be found in Lahuis et al. [7] Figure 17 shows an example from this paper. In cases where the extent of the molecular band is too large this may not work and the best results will be the optimised RESP_INTER corrected data. An example of this is shown in section 15.3.2.2.

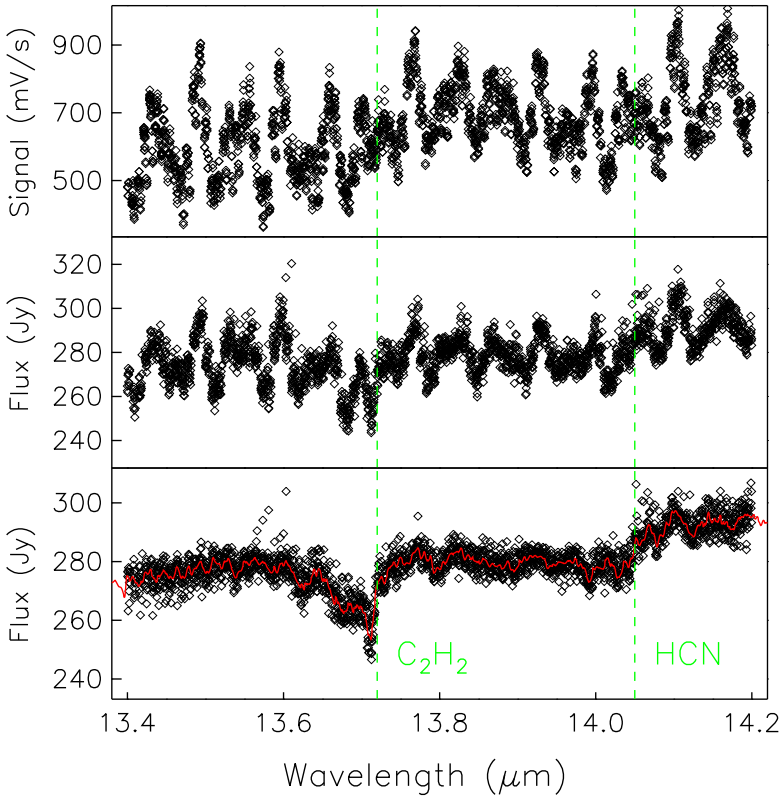

Figure 17: Example of optimal defringing as described in section 15.4.2.2. In this example we zoomed in on the wavelength range around 14 μm where we find gas-phase C_2H_2 and HCN. The wavelength range used in the defringing was larger, app. from 13 to 16 μm. In the top panel the raw AOT-6 SPD data from one detector is shown. In the middle panel the RSRF corrected data for all detectors and in the lower panel the defringed data very clearly showing the molecular absorption data.

band	aperture	cycle	ncycle
1	all	4000	1000
2	2	4000	1000
2	3	5500	1000
3	1	3000	3000
3	2,3	3000	600

Table 16: Default search range for fringe components for each band in FRINGES.

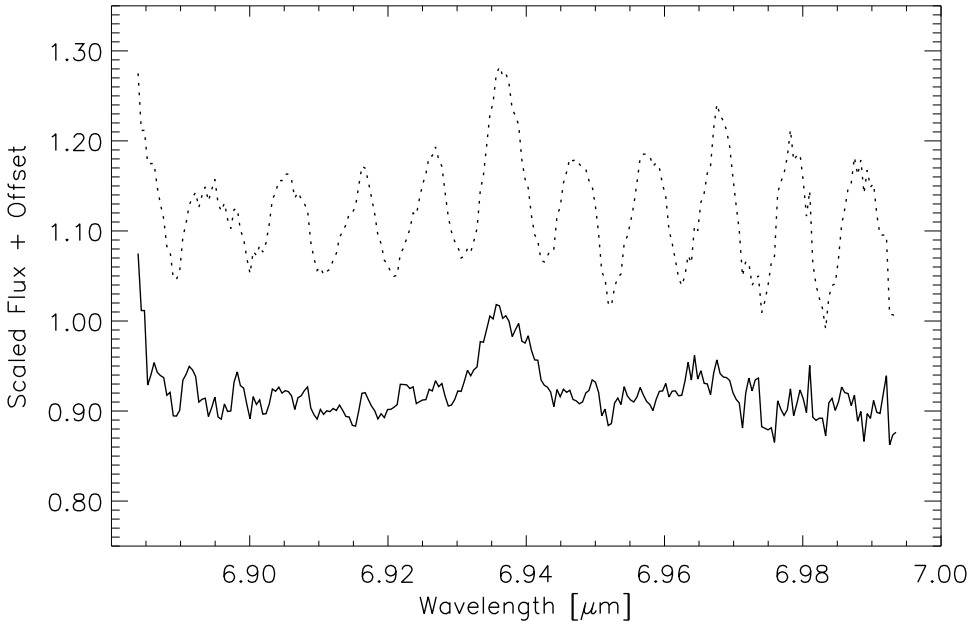

Figure 18: An example of a single SWS02 scan that can be defringed well using AARFRINGE. A relatively narrow line is superposed on a significant number of well-defined fringes.

15.5 AARFRINGE

The last routine available to tackle fringes is AARFRINGE. It is written to in principle only work on rebinned AAR data. It should be noted that it is advised to first try to do the defringing using RESP_INTER and FRINGES. In cases where these fail it may be possible to try defringing using AARFRINGE. The calling syntax is quite simply

```
aarout=aarfringe(aarin)
```

where aarin is the already processed and rebinned input aar. Details of use are described in the OSIA dc-file. The major steps to be taken within AARFRINGE are

- Selecting an AOT band. Fringes are coherent only within AOT bands, so that bands have to be treated separately.
- Masking out outliers that may occur at the edges of the rebinned spectrum, and strong lines. These might fool the fringe location algorithms. The DOK method (see below) is most robust to such problems but even there it is often useful to take out such regions before running the fringe detection.
- Selecting one of the three methods:

  FFT This interpolates over peaks in the Fast Fourier Transform of the data to take out fringes. It works best with long stretches of data like entire AOT bands but often gives imperfect correction at the edges of this range.

  DOK This implements the robust iterative sine fitting algorithm introduced by Do Kester for FRINGES, but working here on the rebinned AAR. Often, it provides the best result. NB. the AARFRINGE implementation of this method is out of date compared with the FRINGES implementation.

  SIN This is a straightforward least-squares sine fit. Because of convergence problems, it is useful only for data with few fringes like single SWS02 scans where FRINGES cannot find proper fringes.

- For all methods, frequency ranges have to be defined in which fringes should be searched for. The OSIA dc file gives more details on this as well as a table which provides some hints concerning known fringe frequencies ranges for the various AOT bands.

15.5.1 AOT-2 scans

In some cases when we e.g. have only a single AOT-2 scan within an AOT band FRINGES does not work optimally because the baseline is just too short. In that case it is possible to achieve good result with the sine fit option in AARFRINGE. Figure 18 shows an example of such a case.

16 HowTo reduce a grating AAR (within the SWS OSIA Environment)

16.1 Introduction

This HowTo document summarises the routines available to the user to manipulate the data in an AAR. This includes e.g. software to remove bad data-points, do flat-field corrections and rebin the AAR data. The intention of this HowTo document is to give a brief description of each routine. A high level description and explanation of the use of keywords and parameters is given in the OSIA dc-documents. For some routines examples are given and some pros and cons are discussed.

We assume here that the user has produced a fully calibrated spectrum in the form of an AAR (*please refer to* HowTo process SWS data from SPD to AAR). Once the user has produced a final spectrum he can start with the scientific analysis.

The use of the following routines is so varied and can be applied in many ways depending on the data or the preference or philosophy of the user that no full example or recipe is presented here. It is advisable for a beginner to first contact an experienced user or one of the data-centers to discuss SWS data reduction. He/She will then quickly come with his own visions on how to use (or not use) the various routines. It is foreseen however that in the following versions of this document some more explicit examples will be presented on the use of some routines and a few examples of 'full' reductions will be described.

16.2 Available Routines

Table 17 lists all available routines within OSIA for the manipulation of AAR data together with a short one line description. A more extended description is given in the paragraphs below. Full descriptions of the routines are given in the OSIA dc-documents.

Module	Purpose
aarfilt	extract a filtered spectrum
aarfringe	defringe a rebinned AAR (*see the HowTo-document* howto-fringes)
cleanstruct	flag or remove unwanted data points
combine	combine two spectra
erase_points	interactively throw out rogue points from an AAR
fringes	defringe a (rebinned) AAR (*see the HowTo-document* howto-fringes)
sigclip	perform sigma clipping
sws_rebin	rebin a spectrum onto a predefined grid or on a grid defined by a constant resolution or constant binsize
sws_grid	create an AAR with a defined wavelength grid
sws_flatfield	flat-field all separate det scans per AOT band, line and scan wrt. each other or a reference spectrum
select	select records of an OSIA structure into a new (shorter) one

Table 17: Major AAR Reduction Routines

16.2.1 aarfilt

aarfilt produces a filtered and regridded output spectrum. The output is either a straight median or a kappa-sigma clipping can be done. The output is sampled on a linear or logarithmic grid depending on the size of the wavelength

region.

16.2.2 cleanstruct

CLEANSTRUCT can be used to remove or mask all sorts of 'unwanted' data. This can be data from a faulty detector or scan, data which is outside the defined AOT-bands or a user preferred wavelength range for a particular band.

The data can either be flagged nodata or mask. In the latter case the data is still considered valid data within OSIA but it will be excluded in particular operations. E.g. sws_flatfield will not use masked data in the derivation of the flat-field parameters, but it does correct them (it will do the same with data flagged as nodata).

Data which is flagged as nodata (e.g. flagged as such by another routine like sigclip) and not wanted in a later stage of the processing can be removed by setting then /nodata keyword.

16.2.3 combine

With combine two OSIA structures of the same type can be merged into one.

```
AAR_COMB = COMBINE( AAR1, AAR2 )
```

16.2.4 select

select is a very useful routine to extract sections of an AAR which can e.g. serve as a reference for flat-fielding. As input it takes the structure and a boolean array defining the data-points to be selected.

```
AAR_SEL = SELECT( AAR, BOOL )
```

Assume we have a deep AOT-2 observation and I do a first inspection of the raw AAR to find the most well behaved scan for a particular line. E.g. for AOT-band 1A (defined by detector number, aperture and order, see the SWS Handbook) I want the data for line 2 and scan 5.
I select this part in the following way:

```
AAR_SEL = SELECT(AAR, AAR.DATA.DET LE 12 AND $
                  TEST_STATUS(AAR,APER=1) AND $
                  TEST_FLAG(AAR.DATA.FLAG,ORDER=4) AND $
                  AAR.DATA.LINE EQ 2 AND $
                  AAR.DATA.SCAN EQ 5 )
```

HowTo understand and manipulate flags / statuswords explains in detail how you can use the status and flag words for detailed data-selection.

The selected part of the AAR can then be used as a reference in e.g. sws_flatfield

```
AAR_FLAT = SWS_FLATFIELD( AAR, 0, REF=AAR_SEL, AOT_BAND='1A', $
                          LINE=2, /OFFSET )
```

This is of course just one example, the possibilities are innumerable, one can select on AOT-band, detector numbers, flux level, wavelength ranges.....

16.2.5 sigclip

Perform sigma-clipping on both aar and spd data. The criteria for clipping can be the deviation of a data-point with respect to the mean or median of the data in a resolution element or the value of the stdev of an individual data-point.

16.2.6 sws_rebin

Rebin a SWS aar spectrum onto a 'regular' grid either defined by a constant resolution or a constant resolution binsize or onto a predefined wavelength grid.

sws_rebin offers a number of options for combining the data within one resolution elements:

FLUXCON flux interpolation over a resolution bin in combination with a Gaussian smoothing/weighting (can be disabled)

MEAN simple average of the data in a resolution bin

WMEAN a weighted average where a combination of weights can be used (*see the dc-document* FLUXCON)

MEDIAN median value for each resolution bin

16.2.7 sws_flatfield

sws_flatfield can be used to correct data which has apparently been affected by some form of instrumental or reduction artifact. This can be e.g. a wrong dark current subtraction as a result of a heavily glitch-affected dark current measurement or clearly visible memory effects.

By default all individual detector scans will be corrected independently with respect to a reference. This can be an internally defined reference (in general the down scan data of all twelve detectors) or a user supplied reference (e.g. a model spectrum or a selected line out of the aar which by judgement of the user best represents the correct flux level).

The correction can be an offset or a gain correction and the data which should be used to derive the correction parameters can be selected interactively.

16.2.8 sws_grid

sws_grid can prepare an aar with a defined wavelength grid based on the physical resolution (as given by the OSIA module resolution) or a grid with a constant resolution.

The thus prepared AAR can then e.g. be used as input to sws_rebin. This can in particular be useful when one is reducing a number of 'similar' observations which ought to be compared. Rebinning the final or intermediate results to one single wavelength grid makes a direct comparison by subtraction or division possible.

16.3 Pitfalls

16.3.1 SIGMA-CLIPPING

Sigma-clipping or removing data on the basis of statistics.

This is always an action which should be executed with care and awareness. Doing e.g. too many clipping action or using too low a resolution or clipping values can result in the removal of real data and thus in the production of erroneous results.

16.3.1.1 AOT-1 Care should be taken when applying this on AOT-1 data, in particular on fast AOT-1 observations. In some cases it is possible to clip off too many data-points and remove real data from emission lines and not only outlying points. The cause of this is that the spectral sampling of AOT-1 data is much lower than that of AOT-2 or AOT-6 data.

16.3.2 FLAT-FIELDING

There are a few obvious dangers with flat-fielding. One can be the degree of flatfielding (e.g. the use of a too high order with too short a baseline), another the choice of reference and the last (at least the last I will mention) the choice of correction (does one apply a drift correction or a gain correction).

16.3.2.1 Flat-field order The choice of the degree of the flat-fielding is important for the desired result. In some cases it suffices to use a low order correction, e.g. with short scans as in an AOT-2. In that case mostly a zero order correction will be sufficient.

However in other cases it is really necessary to use a higher order correction. If one wants to correct e.g. the memory effects often seen at the start of a band 2 or band 4 scan one has to use a higher order correction to correct the affected up scan to the down scan.

16.3.2.2 Reference spectrum The choice of the reference spectrum can be very important for the result of the flat-fielding action.

E.g. `sws_flatfield` will select the down scan data as the default reference data. The assumption here is that this data is least affected by memory effects. However this may not always be so, the down scan can e.g. be affected by band glitches or in the case of band 2 data flux history may play a role, making the down scan more unreliable than the up scan.

Also the resolution of the reference data is important. A low resolution reference spectrum is in principle okay, `BUT` the presence of strong unresolved lines will then become a problem (also with a high resolution reference these can become a problem). In that case one can best block these lines interactively with `sws_flatfield` using the keywords `select` or `mask`. Only the selected and/or the non-masked regions are then used for the derivation of the flatfield parameters, all data is however corrected.

So always be aware of the data which is used as a reference and if necessary select only a part of the data to be used as a reference.

16.3.2.3 Offset or gain What correction should I apply, a gain or an offset correction? The answer to this is easy but at the same time not.

One should apply an offset correction when the drifts are caused by 'dark current' drifts and a gain correction when sensitivity changes become important. When that is for a particular observation may be another story.

The simplest assumption is that when one is dealing with low-signal data one is most affected by offset changes. At high flux levels it is most likely caused by memory effects and sensitivity changes and thus one should apply a gain correction.

A nice example of this could be e.g. a spectrum with a strong continuum and a fully saturated silicate absorption feature. In this case one should do both an offset and a gain correction. How? This can be done with `sws_flatfield` by doing a two step correction. The first step is an offset correction using the fully saturated part of the silicate feature (using

the select or mask keywords) and the second step is a gain correction on the high signal data on both sides of the absorption.

16.4 AOT Specifics

All SWS grating AOT's (AOT-1, AOT-2 and AOT-6) have in common that the observation always has both an up and a down scan. In SWS terminology an up scan means a scan where the SWS grating scanner is scanning in increasing grating position (LVDT). If the grating position is increasing the wavelength is decreasing. So a SWS up scan is a scan with decreasing wavelength. The up scan is always preceding the down scan.

16.4.1 AOT-1

The AOT-1 is an observation covering the full wavelength range of SWS. It is taken at low resolution (a factor of app. 2, 4 or 8 of the full SWS resolution) in order to reduce the observation time.

Between any two AOT-bands there is a region covering the same wavelength. This is called the off-band data. This data is in general less reliable and less sensitive than the in-band data (as defined in the SWS Handbook) but can be used in particular cases. It can e.g. be usefull for the confirmation of features, provided the off-band data has a sufficient SNR.

16.4.2 AOT-2

AOT-2 data are short scans covering only a small wavelength region in general focussed on a spectral line. The line scan can be repeated many times depending on the signal to noise the user requested.

The main disadvantage of all AOT-2 data is the short baseline of the scans with a wavelength coverage changing from detector to detector. This makes corrections more difficult.

16.4.3 AOT-6

AOT-6 data are scans over a limited range from as small as an AOT-2 up to a full scan covering a whole AOT-band. Contrary to AOT-1 data however, AOT-6 data does not contain off-band data.

AOT-6 scans can also be repeated more times to achieve a higher signal to noise ratio.

In general an AOT-6 observation will have a larger baseline than AOT-2 data.

16.4.4 AOT-7 SW grating data

The SW data of an AOT-7 observation are similar to AOT-6 data.

16.5 Future Developments

For the next version of this HowTo explicit examples of indivudual routines are planned and more details and accompaniyng illustrations of AOT specific problem will be given.

If you think any OSIA functionality is missing in this document or if you find descriptions incomplete or inadaquate, please send me an e-mail so that it can be included in the nevt version of this document.

17 HowTo reduce an SWS AAR AOT7

This document describes a way to process SWS AOT07 AARs, the Fabry-Perot part of the data, to a rebinned spectrum.

17.1 Introduction

SWS AOT07 AARs are not different than the products of any other SWS AOT. The AARs can contain grating data in the SWS short-wavelength section and always contain SWS Fabry-Perot data. For handling the grating data please refer to HowTo reduce a grating AAR (within the SWS OSIA Environment) document. This document assumes that a valid and carefully processed AAR structure (variable name "aar") of your observation is available in your IDL OSIA session (see also HowTo process SWS data from SPD to AAR). The valid Fabry-Perot (FP) data is contained in detector 49 and/or 51 depending on the observed wavelength range. Pixel 49 and 51 cover the range between 11.4-26.0 μm and 26.0-44.5 μm respectively (see SWS Handbook for more details).

17.2 Flatfielding

SWS AOT07 observations consist of at least 3 scans of an individual spectral range, the actual number depends on the requested S/N of the specific observation. Check the line number tag of the AAR for example by:

```
plot,aar.data.line,psym=1
```

to see how many scans were executed. Restrict yourself on one spectral line of one FP section only at a time for the processing:

```
ind=where(aar.data.wave gt lambda1 and aar.data.wave lt lambda2 and
          aar.data.det eq det)
aarnew=define_aar(header=aar.header,data=aar.data(ind))
```

The new AAR structure "aarnew" contains all data of the specified line only. Note that all FP scans are in one scan direction only (from large to shorter wavelengths). Continuum values of AOT07 observations have rather low accuracy (absolute and sometimes also in shape), it is therefore recommended to use other grating observations as references for the continuum. If there are no grating observations available, a good approximation is a sort of constant "mean" or "median" value of the whole data:

```
; example
flat=aarnew
flat.data.flux=median(aarnew.data.flux)
```

The new AAR "flat" serves now as reference during the SWS_FLATFIELD procedure:

```
; example
aarnewf=sws_flatfield(aarnew,1,/scaling,ref=flat)
```

Note the different options for SWS_FLATFIELD (see also HowTo reduce a grating AAR).

17.3 Clipping

At this stage outliers can be removed via SIGCLIP. Note that observations with the SWS FP have usually an instrumental resolution of ∼30000, therefore a value of that order has to be passed to the SIGCLIP procedure. A sigma value of 3 has

turned out to be rather reliable, values down to 2.5 can sometimes improve the result further. Low sigma values can end in total removal of all data if stop=0 is specified as well. Consult also the module specific dc-documentation of SIGCLIP.

```
; example
aarnewfs=SIGCLIP(aarnewf,resol=30000,lines=0,stop=0,sigma=3)
```

17.4 Rebinning

The combination of the different scans can best be done via the module `SWS_REBIN`. Again, note the intrinsic resolution of the data and your goal, the resolution you wish to rebin to. Consult also the module specific documentation of `SWS_REBIN`.

```
; example
aarnewfsr=SWS_REBIN(aarnewfs,resol=30000,lines=0,over=6,/nosmooth)
```

The structure "aarnewfsr" is the final output of the data reduction for nominal observations.

17.5 Troubleshooting

In case you encounter problems during above processing, or you are not satisfied with the final result, please contact one of the data centers for further advise. Potential problem cases are too numerous to be discussed here in detail.

18 HowTo reduce data using the LOW_SIGNAL analysis tool

HowTo do low signal data analysis, ..., very carefully!!!

The most important points when reducing low signal data is a carefully controlled analysis combined with a large dose of common sense. This means it is essential to check almost every step of the processing and judge its impact.

Originally low signal data was synonymous to data with weak emission lines on virtually zero continuum. Over time the definition of 'low signal' loosened and it can be weak emission lines on zero or strong continuum or absorption data with a depth of only a few percent. In principle it means any spectral feature which is signal-to-noise limited.

An OSIA tool has been written which is intended to help the user in analysing this type of data. The tool, LOW_SIGNAL offers a number of selection and filtering options to reduce the data plus basic line fitting. The tool was originally designed to analyse 'deep integration' AOT-2 data. Therefore the description of the data and reduction tools is sometimes focussed towards this type of data. It will however work for data from **any** of the SWS observing modes.

Most of the reduction options described here can also be executed on the OSIA command line. The main advantage of using the LOW_SIGNAL tool is that it is highly interactive and one can directly see the effects of the filtering and judge the effect via plots and data statistics.

18.1 Reduction tool

The LOW_SIGNAL tool is designed to help the user in finding the optimal way to analyse his SWS data. The tool has a number of data selection possibilities and reduction options (see section 18.2). Step by step the 'best' combination of data selection and reduction can be found. For this purpose LOW_SIGNAL gives a large amount of statistical information to the user (see section 18.1.2.1). With this way the user can check if what he has done improves e.g. the r.m.s. in the reduced spectrum.

In all cases it is advised to first do a reduction without any selection and filtering. At any time in the reduction it is also wise to sometimes do a few steps back since a combination of filters can lead to unexpected results, e.g. artifacts which have the appearance of a spectral line.

Figure 19 shows the widget tool in full. Shown is the main widget itself with in the background three extra windows. Two plot windows, one to show the original input data and one plot showing the result from the defringing (when selected). The third window is base IDL window in which the statistics is printed.

On the left side of the widget you see the data selection and filtering sections. The data selection to select a particular part of the observation, the selection on detector number, the various filter options and a basic line fitting interface. On the right in the widget the basic control and the main plot windows. In the top plot the final reduced and smoothed spectrum and in the lower plot the data filtering with in the end the reduced data with the clipped data overplotted.

18.1.1 Type of data

LOW_SIGNAL was originally designed for the reduction of deep integration AOT-2 data containing weak unresolved lines on a low continuum. It will now also work properly for other types of data. It will probably not be optimal for all types of data. Future improvements to LOW_SIGNAL can incorporate additional functionality to make it more flexible and suitable for more types of data. Suggestions or wishes would be most appreciated and will certainly be taken into account in future updates of LOW_SIGNAL.

Some peculiarities of SWS data and in particular AOT-2 data are described in Appendix 18.3

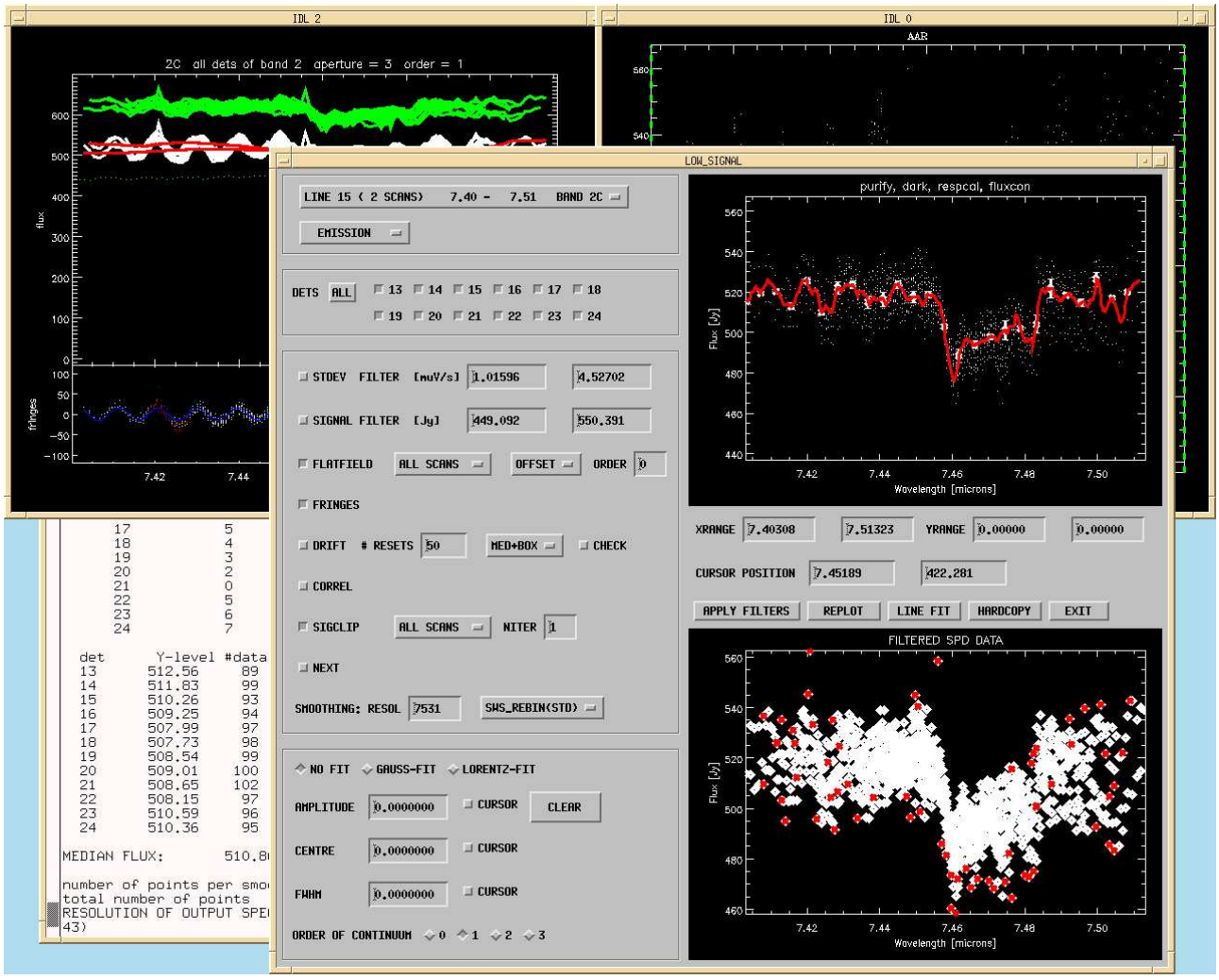

Figure 19: A data reduction session using the Low Signal Analysis tool in OSIA. What we see here are the main components of interest. Above in the back the two basic IDL windows with the uncorrected data and the FRINGES plot output. On the left below we see the IDL command line window where all the user information on selection, filtering, statistics and line fitting is found. In front is the main LOW_SIGNAL widget. On the left in the widget the user control for data selection, filtering and line fitting. On the right part of the widget from top to bottom: the reduced and smoothed data, the selected data range and cursor position, the action buttons to start the processing, line fit or make a hardcopy and finally in the lower plot we see the reduced data with overplot the clipped data. When the 'FIT GAUSS' button is pressed the gauss fit will be overplotted in the top plot and in the lower plot the fit residual will be plotted.

18.1.2 Data input/output

Both SPD and AAR data can be used as input to LOW_SIGNAL. SPD input must be fully processed up-to the VELCOR stage. The document How to process SWS data from SPD to AAR gives all details about the different processing steps.

The call with the possible parameters to LOW_SIGNAL:

```
LOW_SIGNAL, AARSPD, LINES, REB, SPD_CORR = SPD_OUT, AAR_CORR = AAR_OUT
```

AARSPD AAR or SPD calibrated and processed in OSIA

LINES input/output structure containing information on the processing, a.o. the selected data, filtering, line fits, ... When defined on input LOW_SIGNAL will use this. This enables a user to save the reduction session to continue later or rerun in another session.

REB input/output AAR with the reduced and rebinned data. If defined on input the data of a processed line will be overwritten in the AAR or appended if not present.

SPD_OUT Corrected SPD to save or to use for further analysis.

AAR_OUT Corrected AAR to save or to use for further analysis.

18.1.2.1 Information In the IDL window (and thus in the OSIA logfile) information is given about the executed operations as e.g. the selected filters, fits and calculated line intensities. Statistics about the different detectors is given which can be used to decide on the exclusion of a particular detector.

18.2 LOW_SIGNAL selection options

In the following sections we describe all the selection and filtering options available in LOW_SIGNAL. The usual way to start the processing in LOW_SIGNAL is to select a particular line and first do a processing run without any data filtering and examine the basic result and detector statistics. This can usually give you a good idea of the quality of the data.

The order of the description given here is also the order of execution in the program. Whenever a standard OSIA routine is used this is given in brackets in the section title. In that case the user can consult the OSIA dc-document for more details. The data selection is done internal in LOW_SIGNAL, all other filtering is done using the OSIA routine PURIFY.

18.2.1 DATA selection

The data in the SPD is presented as a series of lines which each belong to a unique AOT-band and have a unique wavelength range. Each line can contain a number of scans with data of all detectors, the AOT-band and the number of scans is included in the selection drop-list.

The user can indicate if the spectral feature he is looking for is an emission or absorption feature. This is in particular of interest once the drift filtering (see section 18.2.7) and/or correlated noise filtering (see section 18.2.8) is done. The default is to assume it is an emission feature.

18.2.2 DET, detector selection

By default all detectors are included. With the buttons in the detector selection area all data for a particular detector can be excluded. The user can use the logged statistics to select possible bad detectors. This could be a detector with a significantly higher RMS than the other detectors.

A high RMS of one detector can sometimes be caused by just a few outlying 'glitch' points which can make a good detector look bad. Using the signal clipping can improve this. It can however also be that after clipping all detectors have more or less the same RMS but that for one or two detectors a large fraction of the data is removed. In such a case it often wise to remove that detector.

Pushing ALL will include all detectors and undo previously made detector selections.

18.2.3 STDEV, standard deviation clipping

The standard deviation tag in the SPD can be used for the exclusion of data-points. Data with a high absolute standard deviation or a high relative standard deviation wrt the rest of the data of a detector is discarded. The idea behind this is that any data-point with a high standard deviation is a reset affected by a glitch or some other defect and should therefore be discarded.

The standard deviation is the standard deviation as calculated in the slope fitting in the DSPD (from ERD to SPD) stage.

Two values can be entered for the standard deviation clipping. The first is to remove data which has a standard deviation higher than the average standard deviation of a detector plus the value entered. The second is to remove all data with a standard deviation higher that the value entered.

Information on the stdev is also given in the IDL window.

18.2.4 SIGNAL, signal level clipping

Absolute clipping levels which apply to the signal of the SPD on input. The input data and chosen clip values are always plotted in the IDL 0 plot window. This way the user will always have an overview of the input data and clipping values. In the widget plot this is quite often not possible because one will quickly zoom in on a particular part of the data.

18.2.5 FLATFIELD, flat-fielding (SWS_FLATFIELD)

A simple zero order flat-field can be made. The user can select if he wants this applied to all separate scans or to all for each detector combined. In both cases all data is used as a reference to make sure that in the case the scans are flat-fielded separately, they are flat-fielded to the same level. The user can opt to use an offset or a scaling correction. The last is an option for high flux data where the detection of a weak spectral line is thought to be noise limited.

```
spd = sws_flatfield(spd, order, ref=ref, aot_band=aot_band, /offset [, lines=0])
spd = sws_flatfield(spd, order, ref=ref, aot_band=aot_band, /scaling [,lines=0])
                                ; /scaling or /offset depends on the selection made in the IDL widget
                                ; lines=0 is used when SCANS COMB is selected in the IDL widget
```

The reference is in all cases the (sigclip-ed) data itself with all line and scan numbers set to zero. This ensures that in all cases the flatfielding is done to the same reference frame.

18.2.6 FRINGES, remove fringe residuals (FRINGES)

A standard call to FRINGES is made (please refer to HowTo minimise fringe residuals in SWS data).

```
spd = fringes( spd, band=aot_band )
```

18.2.7 DRIFT, drift filter (DRIFT_FILTER)

A continuum/drift baseline is determined by applying a median filter to the input data. Mini-deglitching (STDEV filtering) is also applied to the data used for the DRIFT filter. The derived baseline is interpolated to all input data and subtracted.

By default a width of 50 resets is chosen. Sometimes it is possible to use a lower value for this (e.g. in the case of AOT-1 data which is already smoothed) , but care should be taken that spectral features of interest are not (partially) removed. In Appendix 18.3 figure 21 it is illustrated why this may happen.

Varying the width of the drift filter and checking what it does to the line profile and line strength will give a good feel for this and should be checked carefully.

18.2.8 CORREL, correlated noise filter (CORR_NOISE)

Correlated noise is used here in the context of noise or events which are common to the whole detector block. This manifests itself as variations of on time-scales of one or two seconds.

Ideally in the absence of spectral detail all detectors would see zero signal. This is obviously not true in a real instrument so what we observe is the zero signal plus some random noise. If the noise is really random the average signal seen by the whole detector block is approximately zero. However quite often we see that this is not so, the average signal of all detectors varies on time-scales of seconds. The most extreme example of correlated noise are the events also known as the 'detector band jumps' where the signal of the detector block makes a sudden large jump.

To correct for this we determine the 'average signal' of the detector block and subtract this. Determining this can sometimes be difficult. The signal seen by the different detectors is the signal at different wavelengths. This means that as soon as there is any spectral detail in the scan the twelve detectors see a different signal. Below it is described how this is done.

18.2.8.1 Application of correlated noise filter When the correlated noise filter is selected by default also the DRIFT filter is activated. The drift filter removes the continuum present in the data and brings all detectors to the same level (with the exception of unresolved spectral lines). The user can switch this of again if he so wishes.

The 'true zero level' is determined by moving a three point mask over the signal of the detector array of one reset. The zero level is assumed to the the average with respect to which the standard deviation is lowest (with exclusion of the three masked points and data flagged as NODATA). The masked points are assumed to be the 'top of the' spectral line or an undetected glitch.

Note however that when there is spectral structure present in the scan and this is not removed properly or when an artificial continuum has been introduced by e.g. bad drift filtering or flat-fielding the correlated noise filter can sometimes do more harm than good. This can be such that weak lines can partly be removed.

If the correlated noise filter is used properly it can improve the final signal to noise ratio. The achieved improvement will depend on the data and the nature of the particular observation.

18.2.9 SIGCLIP, sigma clipping (SIGCLIP)

SIGCLIP is applied using the Chauvenet criterion. The clipping can be done on all individual scans or on all scans combined.

```
spd = sigclip(spd, /chauvenet, aot_band=aot_band, niter=niter [,lines=0])
                                    ; lines=0 only when SCANS COMB is selected in the IDL widget
```

18.2.10 NEXT filter

The NEXT filter removes all data which follows a data point masked by the STDEV filter or by SIGCLIP.

18.2.11 SMOOTHING

The default smoothing is done by rebinning the data using SWS_REBIN to a resolution approximately. twice the instrumental resolution. This is done so as not to loose any spectral detail.

18.2.12 Line fits (GAUSS_FIT or LORENTZ_FIT)

A simple line fit using a Gaussian or a Lorentzian profile can be make. In most cases this is quite sufficient when a single line can be seen by eye. Sometimes however the fit needs to be guided. In that case the user can click on one of the cursor buttons in the fit section and then with a drag or click of the mouse initialise the amplitude, centre and fwhm.

If no proper fit can be made or if more complicated fits (e.g. with more than one component) need to be made it is advised to use the OSIA module MULTI_FIT or use the AAR ouput and do the line fitting in ISAP.

18.2.13 Lineflux (LINEFLUX)

Linefluxes in $\mathrm{W/m^2}$ are calculated for both the reduced spectrum and for the Gaussian fit. The flux is calculated over a region $\pm 3\sigma$ around the centre wavelength of the fit. The results are printed in the IDL command line terminal.

18.3 Description of a spectral scan

We use a spectral scan in an AOT-2 as the prime example here. The issues described similarly apply to AOT-1 and AOT-6 data but in AOT-2 the problems are usually more severe as a consequence of the short scan baseline.

18.3.1 AOT-2 scans

An AOT-2 scan by nature of its short baseline has a few particularities you should be aware of and understand before starting any data-reduction. Each AOT-2 scan has a set length of app. 100 resets, individual data-points, per detector. 50 in the up-scan and 50 in the down scan, both covering the same wavelength range. The wavelength range covered shifts slightly from detector to detector. A result of this is that the sampling density varies strongly over an AOT-2 scan and consequently also the final SNR of the reduced spectrum, app. a factor of 2 from the centre to the edge of the scan.

Figures 20 and 21 illustrate the coverage of spectral lines at different positions in wavelength.

18.3.2 Resolution

For our purposes we need to be aware of three resolution values:

Spectral Resolution spectral resolution of an unresolved spectral line

LVDT Resolution resolution in LVDT units of an unresolved spectral line

Detector Resolution resolution in detectors of an unresolved spectral line

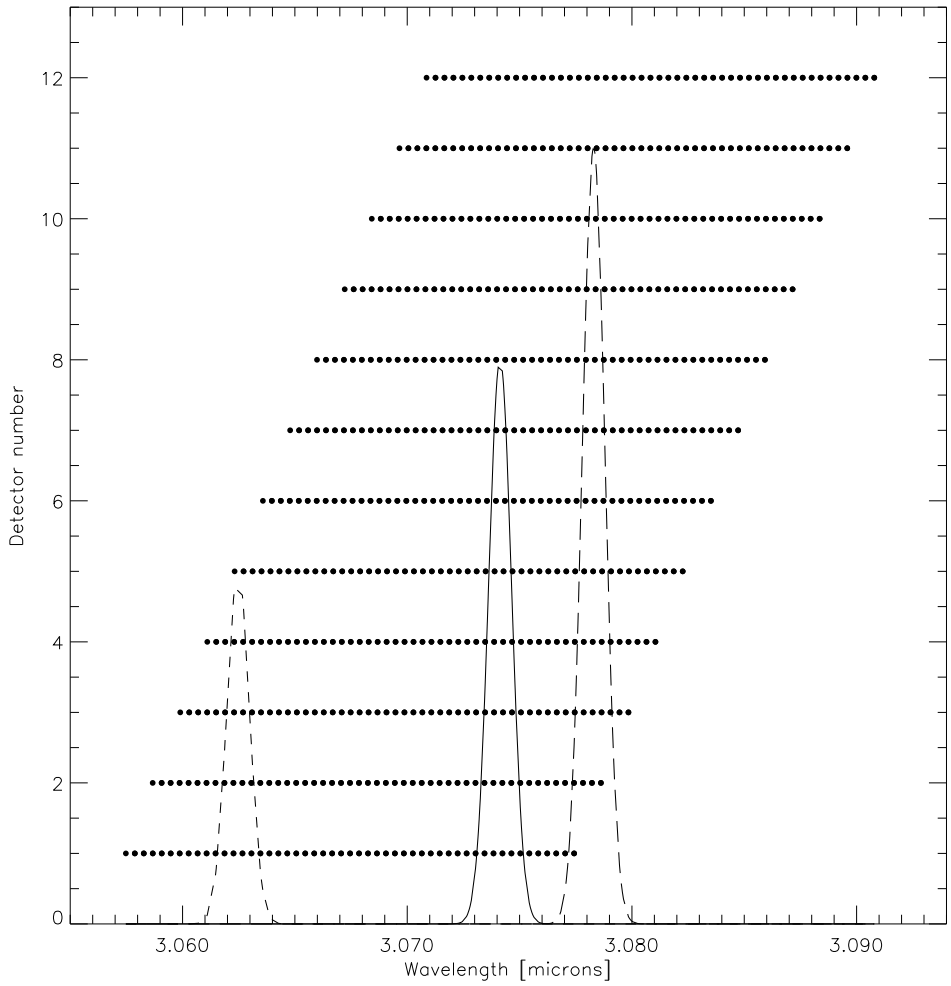

Figure 20: Detector coverage of lines at the edge, centre and slightly off-centre in an AOT-2 scan.

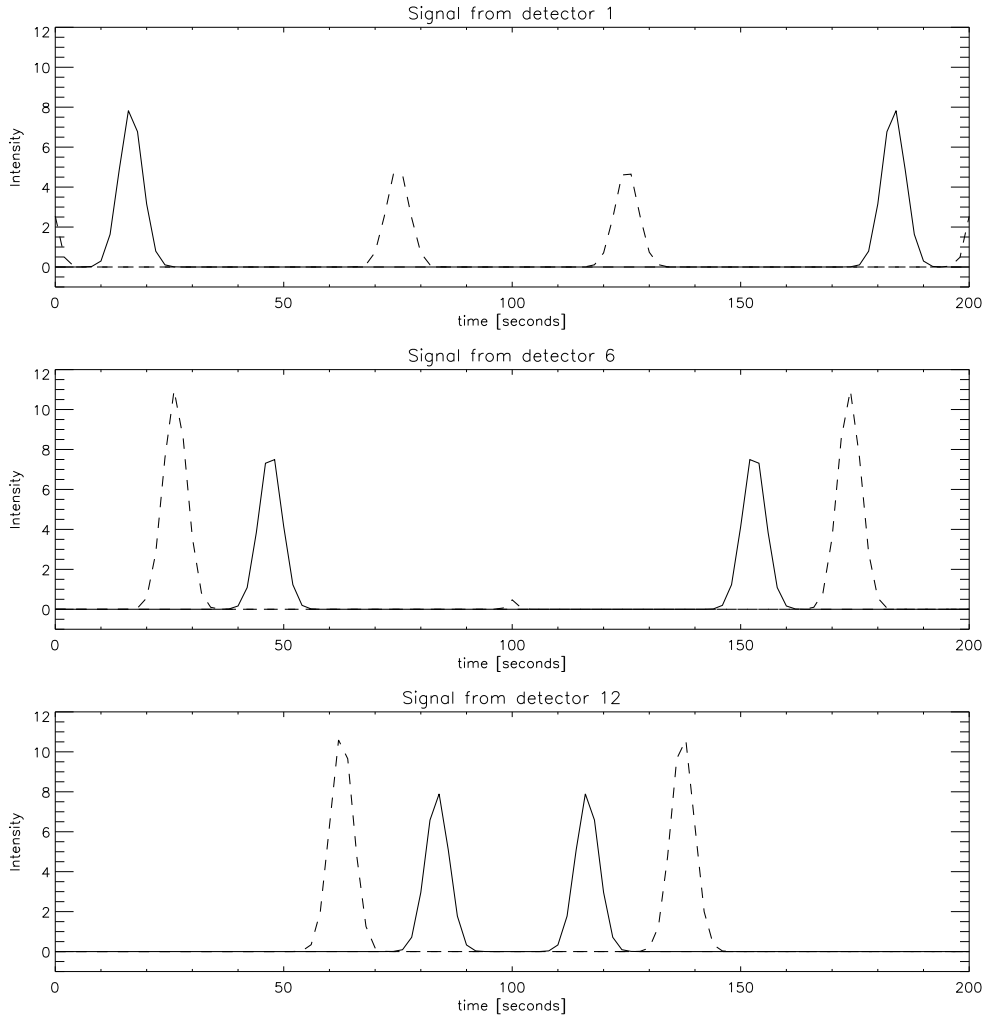

Figure 21: AOT-2 scan with line for detector 1, 6 and 12. The lines drawn are the same as in figure 20.

18.3.2.1 Spectral Resolution The Spectral Resolution of the grating section of SWS is approximately 1000 to 2000 depending on AOT-band and wavelength. The IA^3 module `RESOLUTION` can give these values for both extended sources and point-sources. In the widget info window (and IDL window) `LOW_SIGNAL` prints the SWS resolution. When smoothing the data it is important to smooth to a resolution higher than the actual resolution of the data. Smoothing with a lower resolution will give a smaller r.m.s. in the final spectrum, but will also reduce the amplitude of the line, which for weak lines is not to be advised.

For strongly resolved lines or spectral structure in the continuum there is the problem of the varying detector coverage over the scan. This means there is a 'strong' discrepancy of the SNR over the feature. The AOT-2 was never intended to be used for this purpose, but this does not exclude that a user has AOT-2 observations which show interesting resolved lines or structures. The user should be aware of the limitations of the AOT-2 scans in using this data.

18.3.2.2 LVDT Resolution The fwhm of an unresolved line is approximately 4 to 6 LVDT depending on the wavelength (~ 4.5 for bands 1 and 2, $5-6$ for band 3 and ~ 8 for band 4). The width of the applied drift filter should be

sufficiently higher than this. Be aware that because of the varying spectral coverage of the different detectors a line which for one detector is separated by approximately 50 LVDT between the position in an up and down scan or following scans, is as good as adjacent for another detector doubling its effective lvdt resolution (see figures 20 and 21). This means that for one detector a drift filter with a width of app 20 might be more than sufficient, but that this is too low for another detector. This can result in an uneven filtering for the different detectors and in a faulty line-flux.

18.3.2.3 Detector Resolution When doing a correlated noise correction the detector resolution is important. The detector resolution varies from a fwhm of app. 1.3 at lower wavelengths to 1.8 at the longer wavelengths. An unresolved line is thus seen by roughly three detectors. This is the reason why a three point floating mask is used with the correlated noise filter to determine the zero level of the detector block. See section 18.2.8

19 HowTo obtain high resolution data for AOT1 observations

This HowTo describes the implementation of the SWS AOT1 high resolution mode for the OSIA module DSPD. AOT-1 (and low resolution CUS) observations can be re-reduced with DSPD to create with a higher than nominal resolution. This mode allows an observation to be split into partial ramps to which multiple connected slopes are fitted. For AOT1 observations this increases the resolution at the expense of the S/N ratio.

A detailed description of this improvement to DSPD is given in [6].

19.1 Introduction

AOT SWS01 makes full SWS grating wavelength scans at a reduced spectral resolution. The scans are effectively continuous, using small and frequent steps, much shorter than the selected reset interval.

19.2 Implementation

An additional keyword 'subslope' has been added to the call of the OSIA module DSPD. This keyword indicates the number of partial ramps which will be computed for one second. The number of partial ramps may be 1, 2, 4 or even 8 subslopes per sec. Note that for 8 subslopes memory problems may be encountered as it increases the SPD size by almost this factor. Please refer to the dc-document `dspd` for further information and to [6].

19.3 Examples

The tool has been extensively tested against several datasets. Some examples of the results from AOT1 high resolution proceesing mode can be found in [6]. A more extensive analysis of using this tool in the reduction of all SWS AOT-1 with speeds 3 and 4 is given in [2], the reduced products of which are publically available in the ISO Data Archive.

Part III

Pointing

20 HowTo determine the size/orientation of the SWS slit on the sky

This HowTo document describes the angle subtended by the detectors on the sky and how to determine the orientation of the slit on the sky.

20.1 Introduction

Section 20.2 gives the definition of the apertures. Section 20.3 describes the angle subtended by the detectors on the sky and how to determine the orientation of the slit on the sky.

While section 20.4 indicates how the slit size and pointing can affect the data, users should read the SWS Handbook (*see the following URL* http://www.iso.vilspa.esa.es/users/expl_lib/SWS_top.html) for more information on such effects.

20.2 Aperture definition

SWS has three different apertures (aperture 4, used by the LW FP, is a virtual aperture offset slightly from aperture 3 to increase the amount of light falling onto the FPs and hence improve efficiency). To observe a target, the spacecraft pointing has to be adjusted so that the target is imaged onto the selected aperture, with a shutter system allowing the selection of one aperture, while closing off the other two.

Each aperture is used for two wavelength ranges, one for the short-wavelength (SW) section of the spectrometer and one for the long-wavelength (LW) section. Since those two sections are otherwise independent, two wavelength ranges can be observed simultaneously.

The edges of the apertures are oriented along the spacecraft y- and z-axis (see figure 22).

y-axis parallel to the spacecraft y-axis, is the non-dispersion, or cross-dispersion, direction.

z-axis parallel to the spacecraft z-axis, is the dispersion direction.

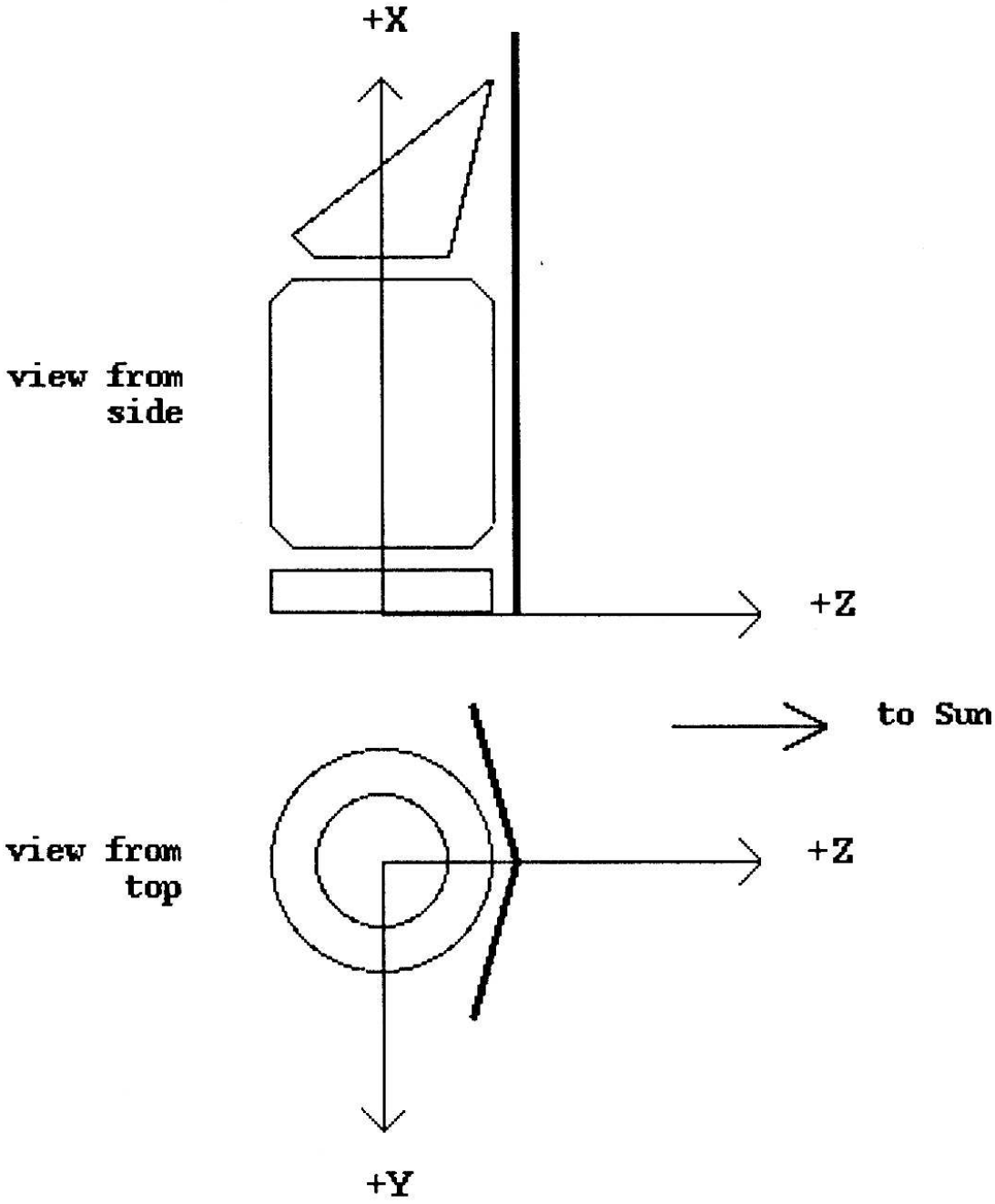

Figure 22: Definition of the spacecraft axes

20.3 Slit Sizes & Offsets

Along the y-axis, the effective size of the aperture is determined by the projection of the detector array on the sky. This amounts to 20, 27 or 33 arcsec for bands 1A to 4, and from 39 to 40 arcsec for the FP bands 5A to 6.

Along the z-axis, the aperture size is determined by entrance slit width projected on the sky (14 or 20 arcsec) for bands 1A to 4. For the FP bands 5A to 6, the aperture is effectively as wide as a detector image on the sky, i.e. either 10 or 17 arcsec.

The aperture numbers, along with their nominal sizes when subtended on the sky, are given in Table 18.

Table 18: Aperture size definition for the SWS AOT bands

	band	Wavelength μm	Detector number	Aperture number	area[3] (z″×y″)
SW[1]	1A	2.38 - 2.60	1 - 12	1	14–20
SW	1B	2.60 - 3.02	1 - 12	1	14–20
SW	1D	3.02 - 3.52	1 - 12	2	14–20
SW	1E	3.52 - 4.08	1 - 12	2	14–20
SW	2A	4.08 - 5.30	13 - 24	2	14–20
SW	2B	5.30 - 7.00	13 - 24	2	14–20
SW	2C	7.00 - 12.0	13 - 24	3	14–20
LW[2]	3A	12.0 - 16.5	25 - 36	1	14–27
LW	3C	16.5 - 19.5	25 - 36	2	14–27
LW	3D	19.5 - 27.5	25 - 36	2	14–27
LW	3E	27.5 - 29.0	25 - 36	3	20–27
LW	4	29.0 - 45.2	37 - 48	3	20–33
F-P1	5A	11.4 - 12.2	49 - 50[4]	1	10–39
F-P1	5B	12.2 - 16.0	49 - 50[4]	1	10–39
F-P1	5C	16.0 - 19.0	49 - 50[4]	2	10–39
F-P1	5D	19.0 - 26.0	49 - 50[4]	2	10–39
F-P2	6	26.0 - 44.5	51 - 52[5]	3[6]	17–40

Notes:

1. SW = short-wavelength grating region

2. LW = long-wavelength grating region

3. 'Aperture area' refers to the dimensions of the SWS entrance apertures projected on the sky.

4. Detector 49 is used for FP observations in band 5.

5. Detector 51 is used for FP observations in band 6.

6. These observations are made through the virtual aperture 4 but are flagged in the data as being through aperture 3.

The actual beam profiles are described in the SWS Handbook.

To derive the position angle of an aperture information on the spacecrafts position on the sky must be used. This can be found in the header keywords CINSTRA, CINSTDEC, CINSTROLL found, for example, in the AAR. Specifically, CINSTROLL is the angle, in degrees, measured anticlockwise, between north and the spacecraft z-axis (see the ISO Handbook for a description of these keywords). Because the SWS apertures are longest along the y-axis we have $PA(SWS) = roll\ angle\ +\ 90^{\circ}$

As an example, if CINSTROLL for a band 1 observation was zero then the 14 arcsec z-axis would be oriented North-South, and the 20 arcsec y-axis would be oriented East-West. The centre of the aperture would be given by CINSTRA and CINSTDEC.

It is possible to use the OSIA commands **LIST_HEADER** and **APER_PLOT** (*see the dc-document* aper_plot) to obtain the CINSTRA, CINSTDEC and CINSTROLL information. **LIST_HEADER** prints out the header information while **APER_PLOT** generates a plot of the selected aperture. Examples of their use are:

```
OSIA> list_header,aar
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                    8 / number of bits per data pixel
NAXIS   =                    0 / number of data axes
EXTEND  =                    T / FITS dataset may contain extensions
COMMENT   FITS (Flexible Image Transport System) format defined in Astronomy and
COMMENT   Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.
COMMENT   Contact the NASA Science Office of Standards and Technology for the
COMMENT   FITS Definition document #100 and other FITS information.
ORIGIN  = 'ESA     '           / European Space Agency
DATE    = '2002-01-04'         / Creation date 2002/004
TELESCOP= 'ISO     '           / Infrared Space Observatory
INSTRUME= 'SWS     '           / Instrument used
COMMENT     SWS Auto Analysis Data
FILENAME= 'SWAA60900804'       / File name
FILEVERS= '2711    '           / Version ID (derived from creation date)
OLPVERS = 'OLP_101 '           / SOC OLP system version
CALGVERS= 'CALG_71 '           / SOC OLP CAL-G files version
USERNAME= 'PIPELINE'           /
OBJECT  = 'GAMMA CRU'          / Target ID as given by proposer
OBSERVER= 'CWAELKEN'           / Proposer ID in ISO Mission DB
EQUINOX =            2000.0000 / Equinox
TMRATE  =                   32 / Telemetry rate in Kbps (Kbits/sec)
EOHAUTCS= '97198033240'        / Approx. UTC of start of observation
EOHAUTCE= '97198052138'        / Approx. UTC of end of observation
EOHAAOTN= 'S01     '           / AOT name
EOHAPLID= 'STARMOD '           / Proposal ID
EOHAOSN = '04      '           / Observation sequence number
EOHAPSN = '00      '           / Pointing sequence number
EOHAPCAT= '4       '           / Proposal category
EOHACIND= '        '           / Calibration indicator
EOHATTYP= '003     '           / Target type
ATTUTCSL= '97198033027'        / UTC of start time of slew to intended target
ATTUTCS = '97198033217'        / UTC of time of first arrival at intended target
ATTOTFTH=            10.000000 / On-target flag threshold (arc secs)
ATTRA   =            187.79121 / Intended Right Ascension of instrument viewing
ATTDEC  =           -57.113030 / Intended DEClination (with ATTRA)
ATTTYPE = 'P       '           / Type of attitude operation (P/R/T)
ATTGUIDE=                68621 / Guide star reference number
ATTSAANG=            98.100000 / Solar aspect angle (degrees)
ATTERROR=                    0 / CONTINGEncy flag(0=success; 1=target not acq'd)
TREFUTC1=            269494360 / UTC (whole seconds since 01-01-1989)
TREFUTC2=              7910510 / UTC (remaining fraction of second)
TREFUTK =           1294232640 / ISO Uniform Time Key (UTK)
TREFITK =           1294232640 / ISO INSTRUMENT Time Key (ITK)
TREFITKU=          0.041666667 / ITK unit length in seconds
TREFCOR1=            269494360 / UTC of 1st reference time
TREFPHA1=            609.38709 / Orbital phase at TREFCOR1
TREFHEL1=            71.500000 / Heliocentric correction +(s) at TREFCOR1
```

```
TREFDOP1=           -19.850000 / ISO velocity towards target (km/s) at TREFCOR1
TREFCOR2=            269497629 / UTC of 2nd reference time
TREFPHA2=            609.42502 / Orbital phase at TREFCOR2
TREFHEL2=            71.290000 / Heliocentric correction +(s) at TREFCOR2
TREFDOP2=           -19.770000 / ISO velocity towards target (km/s) at TREFCOR2
TREFCOR3=            269500898 / UTC of 3rd reference time
TREFPHA3=            609.46294 / Orbital phase at TREFCOR3
TREFHEL3=            71.090000 / Heliocentric correction +(s) at TREFCOR3
TREFDOP3=           -19.690000 / ISO velocity towards target (km/s) at TREFCOR3
ISRRSEV =                    1 / RTA maximum severity level = WA
ISRQSEV =                    1 / QLA maximum severity level = WA
ISRRWARN= 'OK       '          / RTA warning flag from IS user
ISRNOOSL=                    0 / No. of out-of-soft-limit errors
ISRNOOHL=                    0 / No. of out-of-hard-limit errors
ISRNMW  =                    0 / No. of monitor warnings
ISRNCVW =                    0 / No. of command verification errors
ISRNBTW =                    0 / No. of bad telemetry errors
ISRNMVW =                    0 / No. of memory verification errors
ISRNSQLA=                    0 / No. of severe QLA errors
INSTRA  =            187.79121 / Reference instrument J2000 right ascension (deg
INSTDEC =           -57.113020 / Reference instrument J2000 declination (deg)
INSTROLL=            296.92000 / Reference instrument J2000 roll angle (deg)
CINSTRA =            187.79121 / Corrected reference instrument J2000 right asce
CINSTDEC=           -57.113060 / Corrected reference instrument J2000 declinatio
CINSTROL=            296.92000 / Corrected reference instrument J2000 roll angle
....
```

and

```
OSIA> aper_plot, aar, '3A'

   RA,DEC Coordinates (in degrees) of the four slit corners:
   RA(0), DEC(0) =        187.79128      -57.117283
   RA(1), DEC(1) =        187.79753      -57.110596
   RA(2), DEC(2) =        187.79115      -57.108833
   RA(3), DEC(3) =        187.78490      -57.115520
```

With an example plot being shown in figure 23.

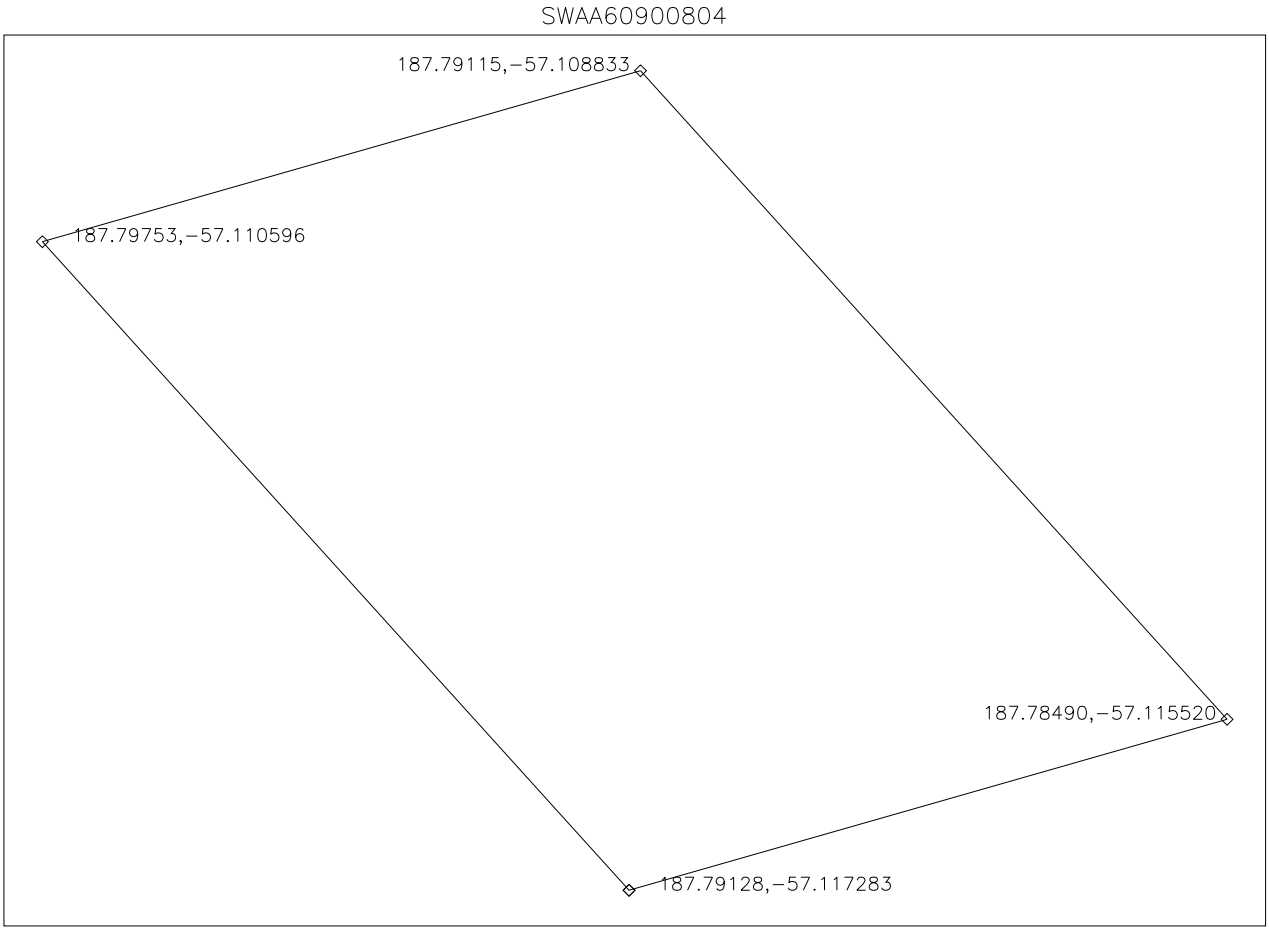

Figure 23: Example output from **APER_PLOT**

20.4 How slit size/pointing can affect observations

The different slit sizes obviously allow different amounts of flux through when observing extended objects, an effect that should be corrected for by the observer.

A more subtle effect can occur if an object has been observed over several months, and an example of how observations of extended objects can be affected is given in the note 'Status of study of AOT02 line profiles and fluxes on NGC6543', by H. Feuchtgruber issued 2 June 1998. In this case the observed line fluxes varied by 10% over the course of several months. This was attributed to the spatial extent of NGC 6543, the SWS slit size projected on the sky and the change in roll angle between the observations.

Information on how the spacecraft pointing can affect observations is given in the SWS Handbook (*see the following URL* http://www.iso.vilspa.esa.es/users/expl_lib/SWS_top.html).

21 HowTo correct mispointed observations and Solar System Objects observations

This HowTo describes how to assess the impact on the SWS data caused by a pointing not centered on the target.

21.1 Introduction

In the ISO SWS Observer's Manual [10], the SWS geometry has a rectangular apertures, with dimensions depending on the selected AOT band. Since the aperture is located in the telescope focal plane, the SWS beam profile was then assumed to be the convolution of this rectangle with the telescope's point spread function.

In reality, the situation is more complex. Within the SWS, the aperture is re-imaged twice. Vignetting and diffraction occur both at the focal-plane aperture and at an internal pupil image. These effects influence the beam profile in the dispersion direction: they make the profile narrower than what should be expected from the slit width. In the other dimension the beam profile is affected by geometrical imaging defects. Both these effects were observed in practice, while the instrument was in-orbit. The profiles were found to be narrower than expected and, on top of that, misalignments between the six detector arrays shifted the profiles to significantly different cross-dispersion positions. These effects made the SWS very sensitive, even more sensitive, to small pointing offsets than foreseen in the original design. This triggered an improvement of the ISO pointing to a level far beyond the specifications [13]. However, the ISO pointing still is a limiting factor in the SWS flux and wavelength accuracy [11].

21.2 The beam profile calibration files

Two calibration files were built based on a theoretical model and the in-orbit beam profile calibrations ([1], [14]), each of them with a different purpose:

- the CAL35 files, with a sampling of $0.1''$ and an extension of $\pm 10'$, is meant for corrections of small mispointing and pointing jitter, used by the routines described in this document, and
- the CAL36 files, with a sampling of $1.0''$ in a range of $\pm 6'$, is meant for extended source correction.

A calibration file has been derived for each AOT band, at the same key wavelengths used in the flux calibration [11]. The beam profiles have been seen to not vary considerably within an AOT band. Only exception are bands 2C and 4, covering a wider wavelength range. For these bands there are then two files provided, at two distinct wavelengths, close to the AOT band borders, also coinciding with the key wavelengths used in the flux calibration.

21.3 The routine `POINT_CORR`

The routine `POINT_CORR` corrects the flux associated to each data point of an SWS SPD file by an amount given by the beam profile calibration file CAL35, for a given spatial offset. By default, the offset between the commanded and reconstructed coordinates are taken, and the user has the possibility to input extra offsets, as determined for instance by a better knowledge of the source coordinates. AOT band dependent offsets can also be given.

The routine works on the time sampling given by the IIPH pointing file, i.e. 0.5 seconds, therefore allowing for a reduction of the beam profile jitter induced noise.

The same tool can directly be used on Solar System Objects (SSO's) observations performed in tracked mode [12], making use of ISO-centric ephemerides, obtained from the JPL facility HORIZON[18] and stored in the OSIA directory `data`. This

[18] (*see the following URL* `http://ssd.jpl.nasa.gov/horizons.html`)

is done via the separate routine SSO_TRACK.

For each data point, the fluxes are divided by the beam profile value correspondent to the offsets between the reconstructed coordinates present in the IIPH file and the intended pointing, as read from the SPD header, or, for an SSO observation, from the ephemerides.

An application of this tool can be seen in Figure24, where the removal of the saw-tooth pattern induced by the tracking is evident, with the tool used in default mode.

21.4 Limitations

The narrow beam profiles and the internal band misalignments make the SWS very sensitive to small pointing offsets, even at the level of one arcsecond. A default use of this routine is therefore not recommended. As described in [13], the ISO pointing can be reconstructed to a level of 1.5 arcseconds, 2 sigma, half cone. The target coordinates (and IR spatial extent) were often not known by the observer with an accuracy of one arcsecond or better. The internal misalignments have been derived to be in the range of 0 to several arcseconds.

The tool should instead be used to assess the effect of the pointing on the data, by tuning the additional, AOT-band dependent offsets.

For details on the tool usage, (*see the dc-document* point_corr) and, for Solar System Objects, (*see the dc-document* sso_track).

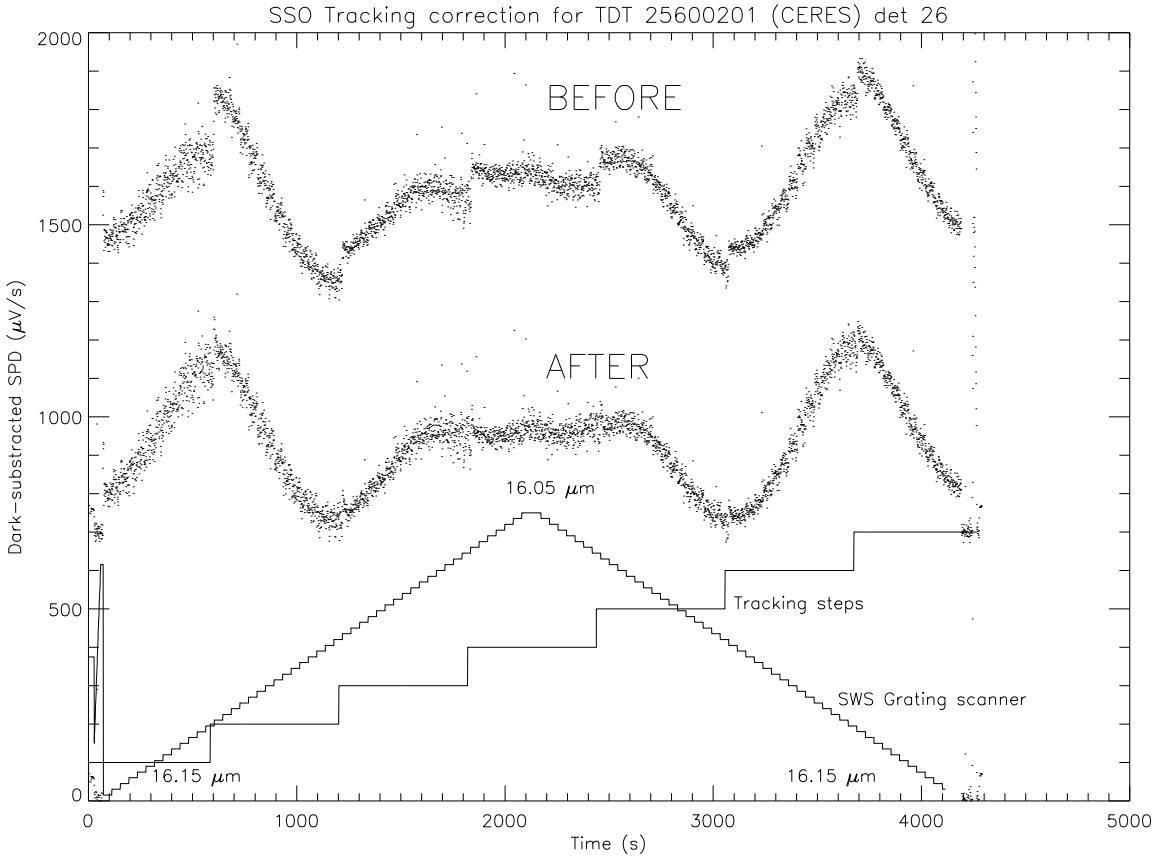

Figure 24: Example of tracking correction on SWS data, an SWS AOT 6 on Ceres, Band 3 detector 26. The offsets between the ISO-centric HORIZON ephemerides and the IIPH reconstructed coordinates for this observation amount to up to $4''$. The detector data show the effects of the tracking while Ceres "moves" on a steep part of the SWS beam profile. The fine staircase pattern illustrates the grating up-down scans. The default use of the tracking correction tool in OSIA removes the tracking pattern almost entirely.

Part IV

Miscellaneous

22 HowTo access SWS errors

This HowTo describes how the user can access errors in the SWS SPD and AAR data products. In the SWS SPD and AAR the error is not propagated as one single total error. Instead three error components are propagated separately.

22.1 Introduction

Originally, in the design of the SWS data products only one error was envisaged, the total propagated error including all error components both systematic and random errors.

In practice this turned out to be an inconvenient methodology. In many cases the systematic errors were dominant and limited the accessibility of the statistical errors. The statistical errors are often useful in later processing such as using them as weights in rebinning or for histogram clipping in low signal data analysis.

For these reasons it was decided to propagate three separate error components, the statistical error, the systematic offset and gain errors. This meant some manipulation creativity was required since only one error was forsenn in the product design. Section 22.2 describes the encoding of the three separate errors.

A complete description of the error propagation of the SWS pipeline is presented in the ISO-SWS handbook Chapter 7.4 (*see the following URL* http://www.iso.vilspa.esa.es/manuals/HANDBOOK/sws_hb/).

22.2 Errors in SPD and AAR

In the SPD processing the `dspd` slope error is propagated without modification. It is contained in the `stdev` tag of both the SPD and AAR. This means it is directly accessible in later processing. It can in OSIA be converted to Jy by using `convert_stdev`. The units in the product header contain the proper units.

The offset and gain errors are not contained in the SPD itself but are propagated in an error structure which should be passed on from routine to routine in the `daar` processing.

The error structure can be created by hand or when the user calls `dark`.

```
err = define_error( spd )
```

The layout of the error structure is reminiscent of the `spd`. For each detector there is an offset and a gain error at each `itk`.

```
idl err structure  |-----type
                   |--history
                   |---header
                   |-----data(1013)
                                  |-det(52)
                                          |-offset
                                          |---gain
                                  |-itk
```

spare[0]	offset error [Jy]	encoding	decoding
0 1 ... 10	0.0 0.1 ... 1.0	10 * offset	0.1 * spare[0]
11 12 ... 109	2.0 3.0 ... 100.0	9 + offset	spare[0] - 9
110 111 ... 199	110.0 120.0 ... 1000.0	109 + (offset-100) / 10	10 * (spare[0]-109) + 100
200 201 ... 255	1200.0 1400.0 ... 11200.0	199 + (offset-1000) / 200 MAX	200 * (spare[0]-199) + 1000 MAX

Table 19: Look up table for the offset error encoded into the aar spage[0] tag.

```
spd = antimem( spd )
spd = dark( spd, errsturct=err )
......
aar = extract_aar( spd, errstruct=err )
```

The stdev error is copied directly into the AAR. The two systematic errors from the error structure, the offset and gain error, are encoded into the spare tag of the AAR.

22.2.1 Error encoding in the aar

In the AAR the offset and gain errors are encoded into the spare tag. The offset error is encoded into the spare[0] tag and the gain error is put into the spare[1] as a percentage error (upto 255%). Table 19 gives the encoding and decoding of the offset error via a lookup table.

A disadvantage of the encoding is that the offset and gain error become digitized. This is most conspicuous for the offset error.

22.3 Using errors in OSIA

In OSIA the AAR errors can easily be accessed via the encode_aa_error and decode_aa_error routines. encode_aa_error accepts a single float array of either the offset error or an absolute error and will return an encoded error to be copied into spare[0].

decode_aa_error can access all SWS AAR errors and will return an IDL structure with the different error components in absolute values and with a *total* error including all available errors.

```
err = decode_aa_error( aar [, stdev=stdev ] )
```

The default offset and gain error are included in this. The stdev error will be included if the units are first converted to Jy (using convert_stdev) or when the stdev keyword is positive.

22.3.1 use of the aar error in OSIA

Since the current implementation and full error propagation in the pipeline was only properly implemented some time after the mission, only a limited number of routines make full use of the different errors.

Also some of the interactive daar routines like dark_inter and resp_inter do not propagate the errors. This should be considered in the analysis when a reduction combining standard pipeline and OSIA specific routines is made.

The most obvious routine one which should properly propagate the errors is sws_rebin.

22.3.1.1 sws_rebin The rebinned spectrum returned by sws_rebin contains two errors. The first is the data noise on the input spectrum (with the assumed resolution). This is returned in the stdev tag of the rebinned AAR. The second is the formally propagated error which is encoded into the spare[0] tag. This includes the offset and gain error from the spare tag and the stdev (slope-)error of the input AAR if that was first converted to Jy with convert_stdev.

In the error propagation in sws_rebin the assumption that 'all separate error contributions are random' is made. This means that the error in the rebinned spectrum does not properly reflect the absolute and pointing errors. For an estimate of these the user should consult the SWS handbook
(*see the following URL* http://www.iso.vilspa.esa.es/manuals/HANDBOOK/sws_hb).

23 HowTo measure a line flux in IA

This HowTo describes how to measure a line flux from already properly reduced SWS AARs. There are two basic options: Doing a direct integration using the GETLINEFLUX command, and performing a nonlinear least-squares fit using the MULTI_FIT command. How to use both of these tools are described subsequently.

23.1 Introduction

The end product of the 'technical' part of most SWS data reductions is a rebinned, single-valued AAR appropriately oversampled with respect to the instrumental resolution. We will assume below that data have already been processed to this level, and that the user wishes to measure line fluxes to be fed directly into his/her scientific analysis. There is however no absolute need to enforce this level of reduction since the two routines will in principle also work on more raw AAR data (MULTI_FIT even on simple arrays of wavelengths and fluxes), though the fit is susceptible to be strongly affected by outliers.

Sometimes the resulting line flux will depend more on previous reduction steps like defringing than on details of the actual methods of measuring the flux. Also, users should be aware that the clipping and rebinning steps, inherent to SWS data reduction, can potentially affect the derived line fluxes.

This is not relevant for 'normal' situations but may be so for either use of unusal parameters in clipping/rebinning, or if requiring extreme accuracy.

23.2 Direct integration using GETLINEFLUX

This module aims at quickly obtaining line fluxes without any *a-priori* assumptions on line shape. Examples for the calling syntax are

```
lli=getlineflux(aar)
lli1=getlineflux(aar1,xr=[12.7,12.9],velzero=12.8135)
```

The input AAR can be either in units of F_ν (Jy) or F_λ (W cm$^{-2}\mu$m^{-1}), provided this is properly registered in the AAR header.

The major steps are

- Internal conversion to F_λ
- Regions of 'good' continuum to the left and right of the line are identified by mouseclicks, a linear continuum is fitted and subtracted. Note that for faint lines on a curved or noisy continuum, this step may have the biggest influence on the final flux.
- The region over which the line should be integrated is identified by further mouseclicks
- The integration proper of the continuum-subtracted line does not assume equidistant sampling. Effectively, it is integration over a histogram defined by assigning all wavelengths the flux of the AAR point *nearest* in wavelength, or more formally

$$flux = \sum_{i=1}^{n} F_\lambda(i)(\lambda(i+1) - \lambda(i-1))/2)$$

for all i such that $\lambda(i)$ is within requested wavelength range. For the first element $\lambda(i-1)$ is replaced by $\lambda(0)$ and the divisor is increased to 4. A similar method is used for the last element. Such a method works well with properly sampled rebinned data but the results may be affected by extreme outliers.

The result is printed on screen and also output as a lli structure

Details are described in (*see the dc-document* `getlineflux.dc3`).

23.3 Nonlinear least-square fit using `MULTI_FIT`

Under several circumstances, a real least-squares fit can be preferable over the fast direct integration described in the preceding section. The profiles of unresolved SWS lines are well approximated by gaussian (grating) or lorentzian (FP) functions. In such a situation, a least squares fit may provide a better definition of continuum and line than obtained interactively with `GETLINEFLUX`, also curved baselines may be introduced. In the other extreme, `MULTI_FIT` provides the option to measure the fluxes of partially blended lines by simultaneously fitting several line components.

To use `MULTI_FIT` for measuring line fluxes, the AAR first has to be converted to a F_λ scale, and it may be necessary to select a line or wavelength range.

A typical syntax may be

```
aar1=spect_form(aar,'w/cm^2/um')
aar2=select(aar1,aar1.data.line eq 15)
multi_fit,aar2.data.wave,aar2.data.flux
```

Further information regarding `MULTI_FIT` may be found in its dc-document (*see the dc-document* `multi_fit.dc1`) however this provides only top-level information on the calling syntax. Extensive details on how to use `MULTI_FIT` and display the results can be found in the actual `MULTI_FIT` window under the 'help' and 'result_i/o' menus respectively.

Part V

Reference list

References

[1] D. A. Beintema, A. Salama, and R. Lorente. SWS Beam Profile Analysis. In *"The Calibration Legacy of the ISO Mission"*, volume ESA-SP 481, page 215, 2003.

[2] W.F. Frieswijk, R.F. Shipman, and F. Lahuis. SWS AOT-1 High Resolution Processing: Documentation. Iso-sws online document, SRON-Groningen, 2004. http://pma.iso.vilspa.esa.es:8080/hpdp/technical_reports/technote12.pdf.

[3] D. Kester, D. A. Beintema, and D. Lutz. SWS Fringes and Models. In *The Calibration Legacy of the ISO Mission*, volume ESA-SP 481, page 375, 2003.

[4] D.M. Kester. *Memory Effects and their Correction in SWS Si:Ga Detectors*. ISO-SWS online documentation, 2001. http://sws.ster.kuleuven.ac.be/documents/memoryeffect/mem_DK_2001.02.01.pdf.

[5] F. Lahuis, H. Feuchtgruber, H. Golstein, D. Kester, W. Luinge, R. Shipman, and E. Wieprecht. SWS signal capture: from flux to signal. In *"The Calibration Legacy of the ISO Mission"*, volume ESA-SP 481, page 387, 2003.

[6] F. Lahuis and D. Kester. *SWS AOT-1 High Resolution Processing*. ISO-SWS online documentation, 2003. http://www.sron.rug.nl/ freddy/ISO/OSIA_4.0/SWS_AOT-1_High_Resolution.ps.

[7] F. Lahuis and E. F. van Dishoeck. ISO-SWS spectroscopy of gas-phase C_2H_2 and HCN toward massive young stellar objects. *aa*, 355:699–712, March 2000.

[8] F. Lahuis et al. *Pulse shape correction*. ISO-SWS online documentation, 2001. http://sws.ster.kuleuven.ac.be/.

[9] F. Lahuis et al. Deglitching of sws low resolution data. In *"The Calibration Legacy of the ISO Mission"*, volume ESA-SP 481, 2003.

[10] K. Leech, editor. *ISO SWS Observer's Manual*. ISO Explanatory Library, 3.0 edition, 1996. http://www.iso.vilspa.esa.es/manuals/SWS_man/.

[11] K. Leech, D. Kester, R. Shipman, and et al., editors. *The ISO Handbook Volume V: SWS–The Short Wavelength Spectrometer*, volume SAI/2000-008/Dc. ISO Explanatory Library, 2.0 edition, 2003. http://www.iso.vilspa.esa.es/manuals/HANDBOOK/sws_hb/.

[12] K. Leech and A.M.T. Pollock. *The ISO Handbook Volume II: ISO–The Satellite and its Data*, volume SAI-99-082/Dc. ISO Explanatory Library, 1.0 edition, 2000. http://www.iso.vilspa.esa.es/manuals/HANDBOOK/II/iso_hb/.

[13] A. Salama. The iso pointing performance. In *The Calibration Legacy of the ISO Mission*, volume ESA-SP 481, page 37, 2003.

[14] A. Salama et al. Beam profile and pointing working group final report, 2001.

[15] R. Saxton, editor. *ISO Data Product Document*. ISO online documentation, 12 edition, June 2000. http://www.iso.vilspa.esa.es/manuals/IDPD/idpd_v12.ps.gz.

[16] E. Wieprecht, O. H. Bauer, D. A. Beintema, D. Boxhoorn, H. Feuchtgruber, A. M. Heras, R. Huygen, D. Kester, D. Kunze, F. Lahuis, K. Leech, R. Lorente, D. Lutz, P. W. Morris, P. Roelfsema, A. Salama, R. F. Shipman, E. Sturm, N. Sym, E. Valentijn, B. Vandenbussche, and E. Wiezorrek. The ISO SWS Data Analysis Software Systems. In *The Calibration Legacy of the ISO Mission*, volume ESA-SP 481, page 285, 2003.

Part VI

Document change log

Version	Date	Initials	Comment
0.0	23 Nov 1998	EkW	First draft
0.1	02 Dec 1998	RH	First working master document
0.2	18 Dec 1998	RH	Comments of DiK included
1.0	21 Dec 1998	RH	First release with OSIA 1.0
2.0	04 Jan 2001	RH	Release for OSIA 2.0
3.0	08 Mar 2002	RH	Release for OSIA 3.0
4.0	24 Oct 2003	AV	Improvements and release for OSIA 4.0